VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Analýza heterogenních informačních sítí

Analysis of Heterogeneous Information Networks

2019 Bc. Jakub Piško

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Jakub Piško**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Analýza heterogenních informačních sítí
Analysis of Heterogeneous Information Networks

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce je prostudovat oblast analýzy heterogenních informačních sítí a seznámit se s metodami rankování, které se v této oblasti používají. Vybrané metody implementovat, dosažené výsledky interpretovat a vzájemně porovnat.

1. Prostudujte problematiku analýzy heterogenních informačních sítí a rankování v nich.
2. Vyberte datové kolekce, předzpracujte je a vhodně uložte.
3. Implemenujte vybrané algoritmy pro rankování v heterogenních informačních sítích.
4. Experimentujte s implementovanými metodami, interpretujte nalezené ohodnocení vrcholů.
5. Vhodně zvolte reprezentaci získaných výsledků (případně vizualizujte).

Seznam doporučené odborné literatury:

[1] Shi, C., Li, Y., Zhang, J., Sun, Y., & Philip, S. Y. A survey of heterogeneous information network analysis. IEEE Transactions on Knowledge and Data Engineering, 29(1), 17-37 (2017).
[2] Sun, Yizhou, and Jiawei Han. "Mining heterogeneous information networks: a structural analysis approach." ACM SIGKDD Explorations Newsletter 14.2, 20-28 (2013).
[3] Liu, Xiaozhong, et al. "Meta-path-based ranking with pseudo relevance feedback on heterogeneous graph for citation recommendation."Proceedings of the 23rd acm international conference on information and knowledge management. ACM, 2014.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Mgr. Pavla Dráždilová, Ph.D.**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019

doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

V Ostrave 29. apríla 2019 .....

Rád by som sa poďakoval Mgr. Pavle Dráždilovej, Ph.D. za jej odborné rady, dohľad a vedenie tejto práce. Ďalej by som sa chcel poďakovať mojim rodičom za podporu počas celého štúdia a kolegovi Ing. Marošovi Pohančeníkovi za jeho odbornú pomoc.

Abstrakt

Cieľom tejto diplomovej práce je zoznámiť sa a preskúmať metódy, ktoré môžeme použiť na analýzu heterogénnych sietí. Vybrali sme a spracovali vhodnú dátovú sadu, ktorú je možné reprezentovať heterogénnou sieťou. Vytvorili sme nástroj na manipuláciu s dátami, ktorý umožňuje výber podmnožiny dát, ktoré chce užívateľ analyzovať. Preskúmali sme sémantiku meta ciest, ktoré možno z týchto dát vytvoriť. Vhodné meta cesty boli využité na vytvorenie bipartitného grafu alebo homogénnej siete pomocou projekcie. V rámci experimentov bola vybraná časť filmovej databázy IMDb, na ktorú sme použili implementované algoritmy ohodnotenia vrcholov grafu. Tieto algoritmy sú určené na analýzu homogénnych alebo heterogénnych sietí a poskytujú tak odlišný podľad na sieť. Výsledky algoritmov boli vizualizované a interpretované.

Kľúčové slová: analýza heterogénnych sietí, ohodnotenie, dolovanie dát

Abstract

The goal of this master thesis is to familiarize with and explore the methods used for heterogeneous information network analysis. We've selected and processed a suitable dataset, which can be represented by a heterogeneous network. We've created a tool to manipulate the dataset, which will allow a user to select a subset intended for analysis. We've explored meta path semantics that can be created from this data. Suitable meta paths were used to create a bipartite graph or a homogeneous network through projection. A part of movie database IMDb was used as input for the implemented ranking algorithms. These algorithms are used for homogeneous or heterogeneous network analysis and provide a distinct view into the network. Algorithm outputs were visualized and interpreted.

Key Words: heterogeneous network analysis, ranking, data mining

Obsah

Zoznam použitých skratiek a symbolov

API	–	Application Programming Interface
DBLP	–	Digital Bibliography & Library Project
EM	–	Expectation–Maximization
HTML	–	Hypertext Markup Language
IMDb	–	Internet Movie Database
MPAA	–	Motion Picture Association of America
ORM	–	Object-Relational Mapping
SQL	–	Structured Query Language
TSV	–	Tab-Separated Values
UML	–	Unified Modeling Language
WPF	–	Windows Presentation Foundation
WWW	–	World Wide Web

Zoznam obrázkov

Zoznam tabuliek

Zoznam algoritmov

Zoznam výpisov zdrojového kódu

1 Úvod

Žijeme v dobe, kedy je svet viac prepojený ako kedykoľvek v histórii. Väčšina dát, jedincov, skupín a komponentov sú navzájom prepojené alebo sa navzájom ovplyvňujú. Spolu tvoria veľké, hlboko prepojené a sofistikované *informačné siete*. Informačné siete sú všadeprítomné a hrajú dôležitú rolu v modernej informačnej infraštruktúre. Analýzou informačných sietí sa zaoberajú vedci v oblastiach informatiky, fyziky, ekonomiky, biológie a mnohých ďalších.

Výskum v oblasti analýzy informačných sietí sa donedávna zameriaval hlavne na homogénne siete. V týchto sieťach sa nachádza len jeden typ objektu a hrany. Heterogénne siete naopak obsahujú viacero typov objektov a hrán, čo umožňuje lepšiu reprezentáciu hlbokých štruktúr a uchovanie skrytej sémantiky. Na takúto sieť môžeme aplikovať širokú škálu analyzačných postupov a metód. Pri analýze heterogénnej informačnej siete sa zameriame na ohodnotenie (ranking) vrcholov. Tým dosiahneme usporiadanie objektov podľa ich dôležitosti v sieti. Na analýzu heterogénnych informačných sietí sme implementovali aj metódu zhlukovania RankClus, ktorá je založená na ohodnotení vrchololov v dvoch odlišných množinách. Niektoré použité algoritmy sú aplikovateľné iba v homogénnej sieti. Implementovali sme preto aj prevod heterogénnej siete na homogénnu.

V kapitole 2 sa bližšie zoznámime s pojmami, ktoré sa vyskytujú pri práci s informačnými sieťami. Preskúmame rôzne typy a metódy analýzy heterogénnych informačných sietí. Uvedieme príklady dátových sád reálneho sveta, oboznámime sa so sémantikou meta ciest. V kapitole 3 detailne popíšeme implementované algoritmy používané na analýzu sietí. Kapitola 4 sa zaoberá vytvorením filmovej databázy, ktorá tvorí doménu heterogénnej informačnej siete tejto práce. Kapitola 5 popisuje implementáciu vytvoreného programu a ako s ním pracovať. V kapitole 6 sa venujeme experimentom.

Vo filmovej informačnej sieti sme použili značenie objektov vychádzajúce z anglického jazyka: herec (A), krajina (C), režisér (D), žáner (G), jazyk (L), film (M) a rok (Y). Z dôvodu zachovania konzistencie s ostatnými publikáciami sú všetky výpisy zdrojových kódov, pseudokódov a ilustrácií uvedené v anglickom jazyku.

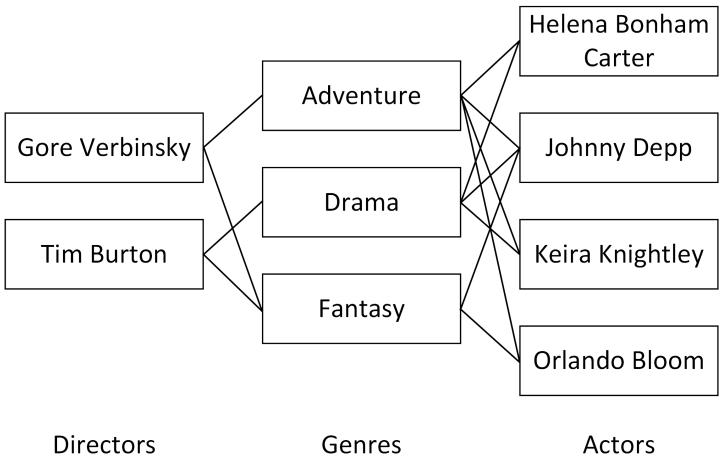
2 Heterogénne informačné siete

Väčšina výskumu v oblasti informačných sietí sa zameriava na homogénne informačné siete, kde uzly reprezentujú objekty rovnakého typu (napríklad osoba) a hrany reprezentujú vzťahy rovnakého typu (napríklad priateľstvo) medzi objektmi. Štúdie v tejto oblasti priniesli mnoho zaujímavých výsledkov, avšak väčšina sietí reálneho sveta je heterogénna, kde objekty (herec, film, režisér) a vzťahy medzi objektmi (hral v, bol režírovaný) majú rôzne typy. Informačná sieť predstavuje abstrakciu reálneho sveta a zachytáva objekty a vzájomné pôsobenie objektov medzi sebou. Ukázalo sa, že takáto abstrakcia je užitočná nielen na reprezentovanie a uloženie základných údajov o reálnom svete, ale navyše poskytuje vhodný prístup k dolovaniu skrytých informácií pomocou skúmania prepojení. Formálne popíšeme informačné siete nasledovnými definíciami [1].

Definícia 1 (Informačná sieť) Informačná sieť je definovaná ako orientovaný graf $G = (V,E)$ s funkciou mapovania objektov $\varphi : V \to \mathcal{A}$ a funkciou mapovania hrán $\psi : E \to \mathcal{R}$. Každý objekt $v \in V$ patrí k jednému kontrétnemu typu objektu z množiny $\mathcal{A}$: $\varphi(e) \in \mathcal{A}$, a každá hrana $e \in E$ patrí konkrétnemu typu relácie v množine relácií $\mathcal{R}$: $\psi(e) \in \mathcal{R}$. Ak dve hrany patria do rovnakého typu relácie, obe hrany zdieľajú rovnaký zdrojový aj cieľový typ objektu.

Na rozdiel od tradičných sietí explicitne rozlišujeme typy objektov a relácií v informačnej sieti. Pre lepšie pochopenie typov objektov a prepojení v komplexnej heterogénnej informačnej sieti potrebujeme popísať jej meta štruktúru.

Definícia 2 (Heterogénna/homogénna informačná sieť) Informačnú sieť nazývame **heterogénna informačná sieť** ak platí $|\mathcal{A}| > 1$ alebo $|\mathcal{R}| > 1$. V opačnom prípade ide o **homogénnu informačnú sieť**.

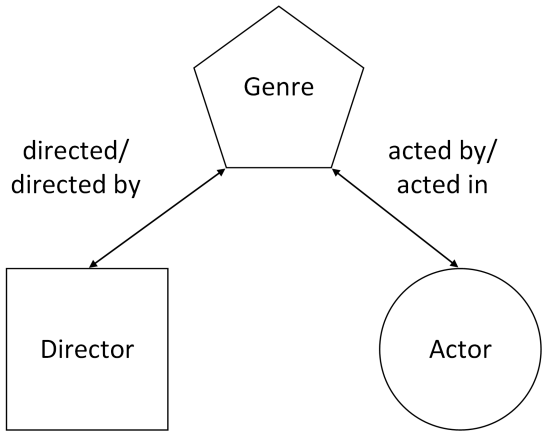


Obr. 1: Inštancia siete

Obrázok 1 zobrazuje ukážku heterogénnej informačnej siete IMDb, ktorú bližšie popíšeme v kapitole 4. Ukážka popisuje 3 typy objektov: režisér, žáner a herec.

Definícia 3 (Schéma siete) Schéma siete $T_G = (\mathcal{A}, \mathcal{R})$ je meta vzor pre informačnú sieť *G = (V,E)* s mapovaním objektov $\varphi : V \to \mathcal{A}$ a mapovaním hrán $\psi : E \to \mathcal{R}$, čo predstavuje orientovaný graf definovaný nad objektmi typu $\mathcal{A}$ s hranami reprezentujúcimi relácie z $\mathcal{R}$.

Schéma heterogénnej informačnej siete špecifikuje reštrikcie na množinách objektov a reláciách medzi objektmi. Tie z nej robia pološtrukturovanú sieť a riadia sémantiku pri hlbšom skúmaní siete. Informačnú sieť, ktorá dodržiava danú schému, nazývame inštanciou schémy. Hranu typu R spájajúcu objekt typu S (zdroj) a objekt typu T (cieľ) značíme ako S $\xrightarrow{R}$ T. Inverzná relácia R^{-1} platí pre $T \xrightarrow{R^{-1}} S$. Vo všeobecnosti sa R nerovná R^{-1}, výnimkou je symetrická relácia R.



Obr. 2: Schéma siete

Obrázok 1 demonštruje reálne objekty a ich prepojenia v databáze IMDb. Obrázok 2 zobrazuje schému siete, ktorá popisuje typy objektov a ich relácií v heterogénnej informačnej sieti.

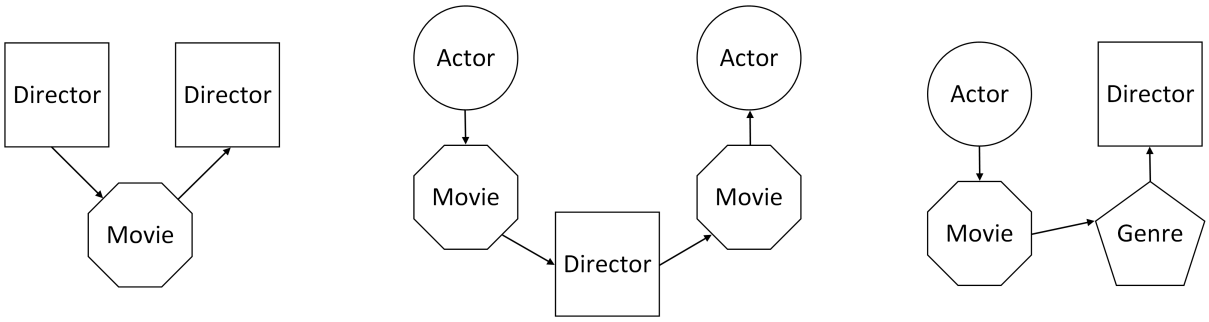
Definícia 4 (Meta cesta) Meta cesta $\mathcal{P}$ je cesta definovaná schémou $S = (\mathcal{A},\mathcal{R})$, značená ako $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \ldots \xrightarrow{R_l} A_{l+1}$ a definuje kompozitnú reláciu $R = R_1 \circ R_2 \circ \ldots \circ R_l$ medzi objektmi $A_1, A_2, \ldots A_{l+1}$, kde $\circ$ značí operátor kompozície relácií.

Pre jednoduchosť môžeme použiť typy objektov na označenie meta cesty v prípade, že množina typov relacií medzi pármi typov objektov $\mathcal{P} = (A_1 A_2 \ldots A_{l+1})$. Ako príklad môžeme uviesť reláciu, kedy režiséri D režírujú filmy M a tú zachytáva meta cesta $D \xrightarrow{\text{režírujú}} M \xrightarrow{\text{sú režírované}} D$, prípadne skrátene *DMD*. Konkrétna cesta $p = (d_1 d_2 \ldots d_{l+1})$ medzi objektmi d_1 a d_{l+1} v sieti G je **inštancia cesty** $\mathcal{P}$ a značíme ju ako $p \in \mathcal{P}$. Meta cesta $\mathcal{P}$ je **symetrická**, ak je aj relácia R definovaná touto cestou symetrická, napríklad *DMD* alebo *AMDMA*. Meta cesty $\mathcal{P}_1 = (A_1 A_2 \ldots A_l)$ a $\mathcal{P}_2 = (B_1 B_2 \ldots B_k)$ sú **zreťaziteľné** iba v prípade, že A_l je rovné B_1 a výslednú zreťazenú cestu označíme ako $\mathcal{P} = (\mathcal{P}_1 \mathcal{P}_2)$. Cesta $\mathcal{P}$ teda označuje $(A_1 A_2 \ldots A_l B_2 \ldots B_k)$. Ako príklad zreťazenia môžeme uviesť zreťazenie ciest *DM* a *MD* do výslednej *DMD*.

Tabuľka 1: Príklady meta ciest a ich sémantického významu v sieti IMDb

Meta cesta	Skratka	Sémantický význam
Režisér - Film - Režisér	DMD	Režiséri d_i a d_j spolupracovali na filme m.
Herec - Film - Žáner - Režisér	AMGD	Herec a a režisér d pracovali na filmoch s rovnakým žánrom.
Herec - Film - Režisér - Film - Herec	AMDMA	Herci a_i a a_j hrali vo filmoch režírovaných rovnakým režisérom.
Herec - Film - Režisér	AMD	Herec a hral vo filme režírovaným režisérom d.
Herec - Film - Žáner - Film - Herec	AMGMA	Herci a_i a a_j hrali vo filmoch s rovnakým žánrom.

V tabuľke 1 sú uvedené príklady meta ciest a ich sémantického významu. Každá meta cesta so sebou nesie iný sémantický význam. Meta cesta *DMD* reprezentuje reláciu režisérov, ktorí spolu režírovali rovnaký film. Meta cesta *AMGD* zase reprezentuje reláciu herca a režiséra, ktorí pracovali na filmoch rovnakého žánru. Sémantický význam meta ciest je dôležitou súčasťou heterogénnych informačných sietí.



Obr. 3: Ukážka meta ciest v sieti IMDb

Výber meta cesty má veľký dopad na úlohy dolovania infomácií. Napríklad miera podobnosti medzi hercami, ktorí sa stretli pri natáčaní filmu v meta ceste *AMA*, bude vyššia ako miera podobnosti v meta ceste *AMDMA*, pretože sa druhá meta cesta zameriava viac na vzťah medzi hercami a režisérmi. Obrázok 3 zobrazuje schému vybraných meta ciest.

2.1 Príklady heterogénnych informačných sietí

Heterogénne informačné siete môžeme vytvoriť z rôznych zdrojov dát. Ako príklad uvedieme niekoľko často skúmaných informačných sietí:

2.1.1 Bibiliografická informačná sieť

Bibliografická informačná sieť DBLP[1] je typická heterogénna sieť, ktorá obsahuje objekty štyroch typov: článok P, konferencia V, autor A a téma T. Každý článok $p \in P$ má väzbu na autorov, ktorí ho napísali, konferenciu, na ktorej bol publikovaný a tému, do ktorej sa radí. Jednotlivé články sú navyše citované inými článkami, čím v sieti vzniká cyklická relácia.

2.1.2 Filmová informačná sieť

Filmovú informačnú sieť predstavuje napríklad IMDb, ktorú bližšie popíšeme v kapitole 4. Objekty vo filmovej informačnej sieti sú napríklad *herec*, *film*, *režisér* a vzťahy medzi objektmi *hral*, *režíroval*, *bol režírovaný*.

2.1.3 Prepravná informačná sieť

Existuje niekoľko podtypov prepravných informačných sietí. Jednou z nich je prepravná sieť mestskej hromadnej dopravy, ktorá obsahuje typy *vozidlo (električka, autobus)*, *trasa*, *zastávka*. Podobnú prepravnú sieť reprezentuje morská nákladná doprava, kde sú typy objektov: *loď*, *trasa*, *prístav*.

2.1.4 Sociálna informačná sieť

Sociálna sieť Twitter[2] tvorí heterogénnu informačnú sieť s objektmi typov *užívateľ*, *tweet*, *hashtag*. Ako príklad relácie medzi objektmi môžeme uviesť: užívateľ *uverejnil* tweet, tweet *obsahuje* hashtag a podobne.

Ako druhú ukážku sociálnej informačnej siete môžeme uviesť webovú stránku na zdieľanie fotografií, Flickr[3]. Táto informačná sieť obsahuje objekty typov *fotografia*, *užívateľ*, *tag*, *skupina* a *komentár*. Príklady relácií zahŕňajú užívateľ *zverejnil* fotografiu, užívateľ *zverejnil* komentár k fotografii, skupina *označila* fotografiu tagom.

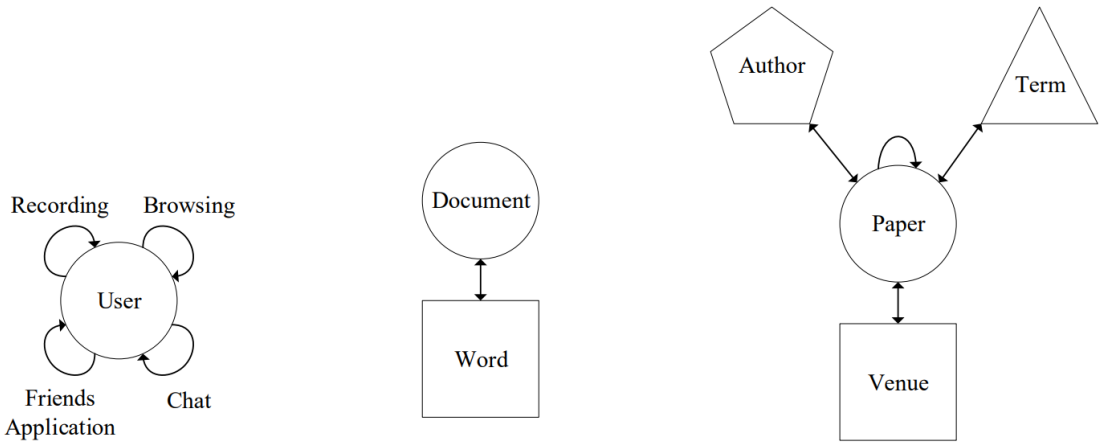
2.1.5 Zdravotnícka informačná sieť

Zdravotnícky systém môžeme reprezentovať zdravotníckou informačnou sieťou, ktorá obsahuje objekty typov *doktor*, *pacient*, *choroba*, *liečba*, *lekárske zariadenie*. Medzi objektmi existujú rôzne typy vzťahov, napríklad pacient *má* chorobu, doktor *lieči* pacientov a iné.

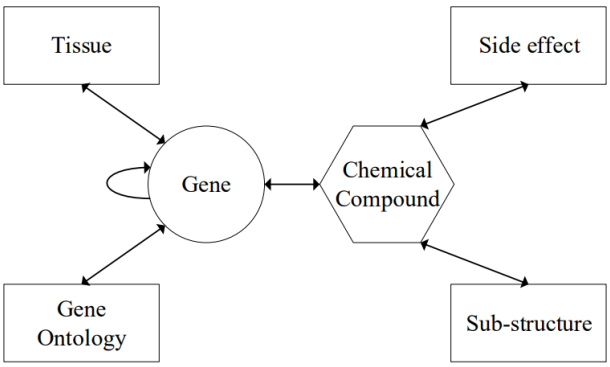
2.2 Typy heterogénnych informačných sietí

Rôzne typy informačných sietí reprezentujeme rôznymi schémami. Obrázok 4 zobrazuje často sa vyskytujúce schémy, ktoré si bližšie popíšeme.

[1] https://www.dblp.uni-tier.de/
[2] https://www.twitter.com/
[3] https://www.flickr.com/



(a) Multirelačná (b) Bipartitná (c) Hviezdicová



(d) Multihub

Obr. 4: Schémy heterogénnych informačných sietí [2]

Multirelačná sieť s jedným typom objektov. Tradičná multirelačná sieť je druh siete, v ktorej sa nachádza jeden typ objektu a niekoľko typov vzťahov medzi objektmi. Tento typ siete sa často vyskytuje na sociálnych webových stránkach, ako napríklad Facebook[4], Twitter alebo Instagram[5]. Obrázok 4(a) zobrazuje schému takýchto sietí [2], v ktorých sú užívatelia navzájom rozsiahlo prepojení.

Bipartitná sieť. Bipartitné siete sú typickým príkladom heterogénnych informačných sietí. Zachytávajú vzťahy medzi dvoma typmi objektov, ako napríklad *užívateľ - položka* [3] alebo *slovo - dokument* [4]. Obrázok 4(b) zobrazuje schému bipartitnej informačnej siete. Rozšírením bipartitných sietí sú k-partitné siete [5], ktoré obsahujú k typov objektov s mnohými väzbami medzi priľahlými typmi objektov.

[4]https://www.facebook.com/
[5]https://www.instagram.com/

Sieť s hviezdicovou schémou. Siete s hviezdicovou schémou sú najpopulárnejšie v oblasti analýzy heterogénnych informačných sietí. Často sú reprezentované databázovými tabuľkami, kde jednotlivé riadky tabuľky predstavujú objekty daného typu a relácie medzi tabuľkami reprezentujú väzby medzi typmi objektov. Je preto jednoduché previesť záznamy z tabuľky do grafu. Takýto graf obsahuje centrálny typ objektu, na ktorý sú ostatné typy objektov priamo naviazané. Takáto sieť je znázornená v obrázku 4(c). Príkladom sietí s hviezdicovou štruktúrou je bibliografická sieť DBLP, kde centrálny typ objetku reprezentuje článok. Ďalšími príkladmi sú filmová sieť IMDb s centrálnym typom objektu film alebo sieť patentov [6] USPTO[6].

Sieť s viacerými centrami. Okrem sietí s hviezdicovou schémou existujú aj komplexnejšie štruktúry, ktoré obsahujú viacero objektových centier. Takéto siete sa často vyskytujú v biomedicínskych dátach [7]. Obrázok 4(d) zobrazuje príklad biomedicínskej siete, ktorá sa skladá z dvoch centier - génu a chemikálie.

Okrem týchto často používaných sieťových schém sa môžeme stretnúť so sieťami reálnych systémov, ktoré reprezentujú omnoho komplexnejšie heterogénne informačné siete. Príklad takéhoto systému môže byť aplikácia, ktorej užívatelia majú vytvorené účty na rôznych sociálnych webových stránkach, s ktorými aplikácia pracuje. V každej sociálnej sieti sú užívatelia prepojení s ostatnými užívateľmi, ako aj s inými typmi objektov: fotografiami, miestami, časovými razítkami a podobne. Vzťahy v takýchto systémoch sú príliš komplikované na to, aby sme ich mohli reprezentovať jednoduchou schémou siete. Príklad reálnej komplexnej schémy siete je graf vedomostí [8].

2.3 Metódy analýzy informačných sietí

V posledných rokoch sa analýza prepojení v grafe posunula výrazne vpred. Boli vyvinuté nové metódy a preskúmané úlohy dolovania dát. Avšak tieto pokroky boli vykonané hlavne v oblasti analýzy homogénnych sietí a týkali sa ohodnotenia, zhlukovania, predikcie hrán a analýzy vzájomného vplyvu. Väčšina týchto metód však nie je jednoducho aplikovateľná na heterogénne informačné siete.

Počiatočné metódy dolovania dát sa sústredili na analýzu vektorov vlastností. V 90. rokoch sa s príchodom WWW objavili nové možnosti v oblasti analýzy prepojení medzi objektmi. Skúmanie sa stále zameriavalo na homogénne informačné siete, konkrétne typ objektov *webová stránka - webová stránka*. Neskôr sa ale výskum zameral na sociálne siete, ktoré obsahujú mnoho navzájom prepojených typov objektov. Je zložité takéto siete modelovať pomocou homogénnych informačných sietí, pričom modelovanie pomocou heterogénnych sietí je prirodzené. S rapídnym nárastom obsahu generovaného užívateľmi týchto sietí sa vynorilo množstvo úloh dolovania informácií, hlavne vo veľkých dátovych sadách. Heterogénne informačné siete sú efektívnym

[6] https://www.uspto.gov/patent

nástrojom na modelovanie veľkého množstva komplexne previazaných dát. Existuje niekoľko kategórií [1] analýzy heterogénnych informačných sietí, ktoré si bližšie popíšeme.

Ohodnotenie. Ohodnotenie v heterogénnych informačných sieťach je dôležitou úlohou dolovania dát, ale prináša niekoľko výziev. Rozdielne typy objektov a relácií so sebou nesú rôzne sémantické významy, ktoré ovplyvňujú výsledky ohodnotenia. Napríklad vo filmovej informačnej sieti získame odlišné ohodnotenie hercov podľa zvolenej meta cesty, keďže meta cesty vytvárajú rozličné štruktúry prepojení medzi typmi objektov. Zároveň sa ohodnotenie odlišných typov objektov navzájom ovplyvňuje, keď existuje predpoklad, že vysoko ohodnotený herec pracuje s vysoko ohodnotenými režisérmi a naopak.

Zhlukovanie. Analýza zhlukov je proces, pri ktorom sieť objektov rozdelíme na menšie podsiete (zhluky) na základe vzájomnej podobnosti medzi objektmi. Málo podobné objekty naopak ostávaju od seba oddelené. Bežné metódy zhlukovania, ako napríklad k-means [9], pracujú s vlastnosťami objektov. V poslednej dobe je často skúmaná oblasť detekcie komunít v sieťových dátach. Tieto metódy modelujú dáta ako homogénnu sieť a používajú na rozdelenie siete metriku normalizovaných rezov [10], modularity [11] a mnoho ďalších.

Klasifikácia. Klasifikácia je úloha analýzy dát, kde zostrojený model alebo klasifikátor predpovedá kategorický štítok triedy objektu. Tradične sa na klasifikáciu používalo strojové učenie, kde rovnako štruktúrované objekty spĺňali požiadavku nezávislej distribúcie. Avšak mnoho prepojení medzi objektmi reálneho sveta túto požiadavku porušuje, čím sa táto metóda stáva zle použiteľnou. Problémy klasifikácie objektov v heterogénnej informačnej sieti majú na rozdiel od tradičných metód niekoľko nových charakteristík. Objekty v heterogénnej informačnej sieti sú rozličných typov, čo nám umožňuje klasifikovať viacero typov objektov zároveň. Štítky triedy objektov nám napomáhajú šíriť vedomosti o objektoch medzi ich rozličnými typmi.

Predpoveď hrán. Predpoveď hrán predstavuje základnú úlohu v dolovaní prepojenia grafu a snaží sa odhadnúť pravdepodobnosť existencie hrany medzi dvoma uzlami na základe pozorovaných prepojení a atribútov uzlov. Predpoveď hrán je často považovaná za úlohu binárnej klasifikácie: pre dva potenciálne prepojené objekty predpovedá triedu, ktorá zodpovedá existujúcej alebo neexistujúcej hrane medzi objektmi. Jeden z prístupov k predpovedi hrán je založený na vlastnostiach štruktúry siete. Liben-Nowell a Kleinberg [12] predstavili prediktory založené na rôznych metrikách vzdialenosti v rámci grafu. Iný typ prístupu využíva na predpoveď hrán informácie o atribútoch. Popescul et al. [13] zaviedli štruktúrovaný logistický regresný model, ktorý na predpoveď existencie hrán využíva vlastnosti relácií.

Meranie podobnosti. Meranie podobnosti vyhodnocuje podobnosť objektov. Je základom mnohých úloh dolovania dát, ako napríklad webové vyhľadávanie, zhlukovanie a odporúčanie produktov. Meranie podobnosti je dobre preskúmaná oblasť analýzy informačných sietí a môžeme ju rozdeliť do dvoch podkategórií: prístup na základe vlastností objektov a prístup na základe

prepojení objektov. Prístup na základe vlastností objektov vyhodnocuje podobnosť objektov na základe hodnôt vlastností objektov, kde počíta kosínusovú podobnosť alebo Euklidovskú vzdialenosť. Prístup na základe prepojení objektov zase analyzuje prepojenia v sieti.

Odporúčania. Odporúčacie systémy pomáhajú spotrebiteľom vytvárať odporúčania produktov, ktoré by mohli užívateľa zaujímať, ako napríklad knihy, filmy alebo reštaurácie. Používajú pri tom široký rozsah techník získavania informácií, štatistiky a strojové učenie za účelom vyhľadania podobných položiek a zákazníckych preferencií. Tradičné odporúčacie systémy pri výbere vhodného odporúčania využívajú primárne užívateľské hodnotenia. Existuje niekoľko spôsobov filtrovania relevantných odporúčaní, ako napríklad faktorizácia matice, ktorá sa ukázala ako efektívny a účinný nástroj v odporúčacích systémoch [14]. Výskum v oblasti odporúčaní založených na heterogénnych informačných sieťach ukázal, že vhodným prístupom k analýze takýchto systémov je skúmanie siete cez meta cesty. Shi et al. [15] implementovali HeteRecom, odporúčací systém založený na sémantike meta cesty pri vyhodnocovaní podobnosti medzi filmami. Systém navyše berie do úvahy aj hodnoty atribútov, modeluje váženú heterogénnu informačnú sieť a navrhuje sémantickú meta cestu založenú na personalizovanej odporúčacej metóde SemRec [16].

Zlúčenie informácií. Zlúčenie informácií označuje proces spájania informácií z heterogénnych zdrojov s rôznymi konceptuálnymi, kontextuálnymi a typografickými reprezentáciami. S nárastom popularity heterogénnych informačných sietí sa spájanie informácií z rôznych informačných zdrojov stalo dôležitou úlohou [17] pri dolovaní dát. Spojením dát z rozličných sietí získavame obsiahle a konzistentné vedomosti o dátovych typoch zdieľaných v týchto sieťach. Zlúčené informácie zahŕňajú štruktúru, vlastnosti a aktivity.

3 Algoritmy pre analýzu heterogénnych sietí

Na podrobnejšiu analýzu heterogénnych informačných sietí použijeme algoritmy popísané v tejto kapitole. Zameriame sa primárne na algoritmy, ktoré ulzy v sieti ohodnotia a umožnia nám tak usporiadať a vizualizovať ich dôležitosť.

Algoritmy ohodnotenia

Algoritmy ohodnotenia (rankovacie algoritmy) nám poskytujú pohľad na sieť, v ktorej určitým spôsobom ohodnotíme a usporiadame uzly grafu podľa ich dôležitosti. Metrika ohodnotenia sa medzi algoritmami líši. Niektoré sa zameriavajú na počet hrán, iné zase na dôležitosť uzlov, ktoré hrany spájajú. Bližšie popíšeme niekoľko implementovaných algoritmov ohodnotenia.

3.1 PageRank

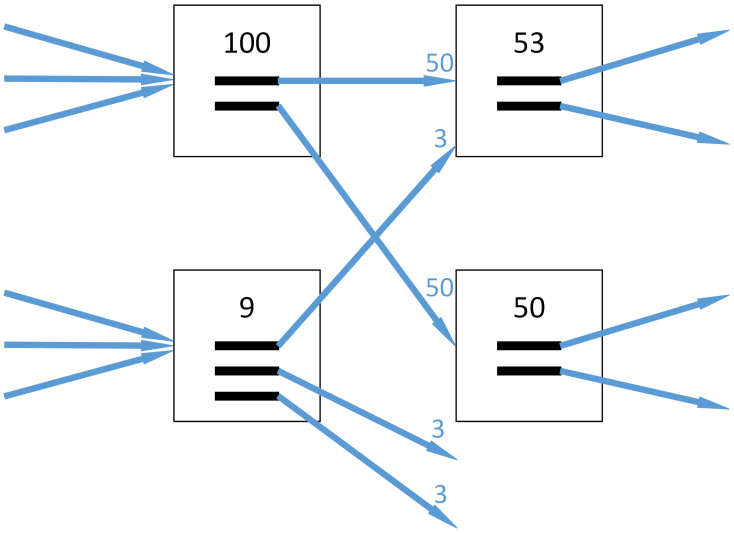
Algoritmus PageRank [18] spracúva hrany grafu a priradzuje uzlom číselnú váhu podľa toho, aké dôležité sú uzly grafu, ktoré daná hrana spája. Pôvodne bol použitý na analýzu prepojenia webových stránok, môžeme ho ale aplikovať na analýzu akéhokoľvek grafu, ktorý reprezentuje objekty odkazujúce na iné objekty, ako napríklad citačná sieť. V roku 1996 ho vyvinuli Larry Page a Sergey Brin na Standfordskej univerzite, kde sa zameriavali na výskum v oblasti webových vyhľadávacích nástrojov. Tento výskum bol motivovaný nedostatočnou presnosťou ohodnotenia webových stránok, podľa ktorého boli zoradené výsledky vyhľadávania. Problém sa rozhodli riešiť vytvorením hierarchie v grafe podľa toho, aké množstvo hrán spájalo daný uzol s ostatnými uzlami grafu. Tým špecifikovali relatívnu dôležitosť uzlu v grafe.

Zjednodušenú verziu algoritmu popíšeme nasledovne na príklade webových stránok. Definujme u ako webovú stránku. F_u reprezentuje množinu webových stránok, na ktoré webová stránka u odkazuje. B_u potom reprezentuje množinu webových stránok ukazujúcich na webovú stránku u. $N_u = |F_u|$ je počet odkazov smerujúcich z u. Výpočet ohodnotenia pre uzol u vypočítame nasledovne:

$$R(u) = \sum_{v \in B_u} \frac{R(v)}{N_v} \ .$$

Treba poznamenať, že ohodnotenie webovej stránky (uzlu grafu) je rovnomerne rozdelené medzi hrany, ktorými je spojená s ďalšími uzlami grafu. Výpočet algoritmu je rekurzívny. Uzlom grafu na začiatku priradíme konštantné ohodnotenie a iterujeme, kým ohodnotenie konverguje. Obrázok 5 demonštruje šírenie ohodnotenia v rámci grafu.

Pri výpočte zjednodušenej verzie však nastáva problém, keď sú dva uzly prepojené medzi sebou, ale neukazujú na žiaden iný uzol zo zvyšku grafu. Pri iterácii algoritmu teda vznikne slučka, v ktorej bude narastať ohodnotenie uzlu, pričom sa nebude ďalej šíriť. Vytvorí sa tým akási pasca,



Obr. 5: Výpočet zjednodušenej verzie algoritmu [18]

ktorá umelo znižuje ohodnotenie zvyšku grafu a negatívne ovplyvňuje kvalitu ohodnotenia grafu. Riešenie tohoto problému spočíva v rozšírení algoritmu o maticu zdroja ohodnotenia $E = \frac{1}{n}ee^T$, kde n je počet vrcholov grafu, $e = (1, \ldots, 1)$ a c je faktor útlmu:

$$R'(u) = c \sum_{v \in B_u} \frac{R'(v)}{N_v} + (1 - c)E \ .$$

Najlepšia hodnota c [19] pre sociálne siete je 0.85.

Prepisom na maticový zápis dostaneme $R = c(AR' + E)$, kde A je štvorcová matica, ktorej riadky a stĺpce korešpondujú s webovými stránkami. Za predpokladu, že $\sum_{u \in B_u} |R'(u)| = 1$, môžeme zápis upraviť na $R' = c(A + E \times \mathbf{1})R'$, kde $\mathbf{1}$ je vektor pozostávajúci zo samých jednotiek. R' teda reprezentuje vlastný vektor $(A + E \times 1)$.

Výpočet algoritmu je relatívne jednoduchý, ak zanedbáme veľkosť dát, nad ktorými ho počítame. Formálne výpočet algorimu zapíšeme nasledovne:

Algoritmus 1: PageRank

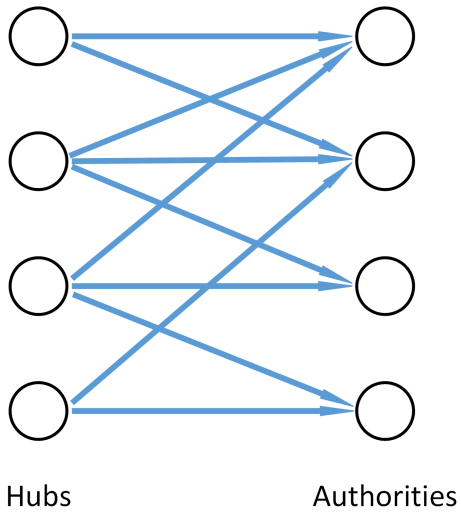
Input: A,ε,c
Output: ranking R
$i = 0$
$R_0 = (\frac{1}{n}, \ldots, \frac{1}{n})$
while $\delta > \varepsilon$ **do**
    $R_{i+1} = c \cdot R_i + (1 - c) \cdot E$
    $\delta = ||R_{i+1} - R_i||_1$
    $i = i + 1$
end
return R_{i+1}

3.2 Hyperlink-Induced Topic Search

Autorom algoritmu HITS [20] je Jon Kleinberg. V roku 1998 sa zaoberal analýzou webových stránok, hlavne v oblasti ohodnotenia výsledkov vyhľadávania. Navrhol spôsob, ako v grafe identifikovať model založený na prepojeniach medzi uzlami. Tento model je založený na vzťahu, ktorý existuje medzi *autoritami* pre danú tému a ostatnými webovými stránkami, ktoré na tieto autority odkazujú - nazval ich *centrá* (hubs). Všimol si, že v grafe (Obrázok 6), ktorý je definovaný štruktúrou prepojení, existuje určitá rovnováha medzi autoritami a centrami. Algoritmus identifikuje oba typy uzlov zároveň.

Akákoľvek množina webových stránok V môže byť reprezentovaná orientovaným grafom $G = (V, E)$, pričom uzly zodpovedajú príslušným webovým stránkam a orientovaná hrana $(p, q) \in E$ naznačuje prítomnosť odkazu z p do q. Výstupný stupeň uzlu p je počet hrán, ktoré z tohto uzlu vystupujú. Vstupný stupeň uzlu p je zase počet hrán, ktoré do neho vstupujú. Maticu susednosti grafu označíme A.

Centrá a autority sa navzájom ovplyvňujú a vzniká medzi nimi vzájomne sa posilňujúci vzťah: dobré centrum je webová stránka, ktorá odkazuje na veľké množstvo dobrých autorít; dobrá autorita je webová stránka, na ktorú odkazuje veľké množstvo dobrých centier. Ak chceme v grafe identifikovať tieto množiny, musíme medzi nimi odstrániť cyklickú závislosť.



Obr. 6: Znázornenie štruktúry centier a autorít

Vzťah medzi centrami a autoritami využijeme v iteratívnom algoritme, ktorý priebežne prepočítava číselnú váhu každého uzlu. Každému uzlu u priradíme pozitívnu *váhu autority* $r_a(u)$ a *váhu centra* $r_c(u)$. Po každom prepočte váhy normalizujeme, aby platilo $\|r_a\|_2 = 1$ a $\|r_c\|_2 = 1$. Predpokladáme, že uzly s vyššou váhou predstavujú lepšiu autoritu, prípadne centrum. Ak p odkazuje na veľké množstvo uzlov s vysokou váhou autority, malo by dostať vysokú váhu centra, to isté platí aj v opačnom prípade. Ak na u odkazuje veľké množstvo uzlov s vysokou váhou centra, malo by dostať vysokú váhu autority. Tento spôsob ohodnotenia definujeme ako operácie

nad váhami uzlu, označené $\mathcal{I}$ a $\mathcal{O}$. Pre váhy $r_a(u)$, $r_c(u)$ zapíšeme operáciu $\mathcal{I}$, ktorá mení váhu autority nasledovne:

$$r_a(u) \leftarrow \sum_{u:(u,v)\in E} r_c(v) \ .$$

Operácia $\mathcal{O}$ mení váhu centra nasledovne:

$$r_c(v) \leftarrow \sum_{u:(v,u)\in E} r_a(v) \ .$$

Z toho vyplýva, že operácie $\mathcal{I}$ a $\mathcal{O}$ predstavujú základný spôsob, ktorým sa autority a centrá medzi sebou navzájom ovplyvňujú. Základná verzia algoritmu vyhľadá rovnováhu hodnôt pre autority a centrá tak, že striedavo aplikuje operácie $\mathcal{I}$ a $\mathcal{O}$ v k iteráciách. Vektor váh $\{r_a\}$ predstavuje vektor a pre každý uzol v grafe G; vektor váh $\{r_c\}$ predstavuje vektor c. Parameter k je počet iterácií algoritmu.

Algoritmus 2: HITS

Input: G,k
Output: ranking a_k, c_k
$z = (1, 1, 1, \ldots, 1)$
$a_0 = z$
$c_0 = z$
for $i \leftarrow 1$ **to** k **do**
  $a'_i = A \cdot c_{i-1}$
  $c'_i = A^T \cdot a'_i$
  $a_i = \text{Normalize}(a'_i)$
  $c_i = \text{Normalize}(c'_i)$
return a_k, c_k

3.3 Chain of Influencers

Algoritmus Chain of Influencers [21] nám umožňuje ohodnotiť heterogénnu informačnú sieť ako reťaz na sebe závislých ovplyvňovateľov. Na jednoduchšie pochopenie algoritmu najskôr predstavíme nasledovné definície:

Definícia 5 (Doména): Doména je množina objektov rôznych typov.

Definícia 6 (Doménová sada): Doménová sada je množina na sebe nezávislých objektov rovnakého typu.

Definícia 7 (Komunitná sada): Komunitná sada je neprázdna podmnožina doménovej sady.

Definícia 8 (Komunita): Komunita je množina komunitných sád, pre ktoré platí:

1. Jedna komunitná sada je zvolená ako bázová sada.
2. Všetky ostatné komunitné sady obsahujú objekty patriace doménovej sade, ktoré sú priľahlé k aspoň jednému objektu z bázovej sady.

Definícia 9 (Ohodnotenie objektu): Ohodnotenie objektu je určenie jeho dôležitosti v rámci komunity.

Definícia 10 (Ohodnotená sada): Ohodnotená sada je komunitná sada, v ktorej má každý objekt priradené ohodnotenie.

Definícia 11 (Reťaz ovplyvňovateľov): Reťaz ovplyvňovateľov je množina aspoň troch za sebou nasledujúcich komunitných sád, pre ktoré platí:

1. Za sebou nasledujúce komunitné sady sú navzájom odlišné.
2. Dve za sebou nasledujúce komunitné sady sú vo vzťahu, kde ohodnotenie objektov komunitnej sady závisí na ohodnotení objektov predošlej komunitnej sady.
3. Prvá a posledná komunitná sada sú rovnaké.

V prípade splnenia týchto podmienok máme možnosť ohodnotiť všetky partity homogénnej projekcie.

Heterogénna informačná sieť predstavuje doménu, ktorá obsahuje objekty rozličných typov a prepojení. Zameriame sa na heterogénne siete, kde sú objekty prepojené rôznymi typmi hrán a objekty v rámci komunitnej sady nie sú prepojené. Takúto sieť môžeme popísať grafom $G = (\{V_1, V_2, \ldots, V_n\}, \{E_{1,2}, E_{2,3}, \ldots, E_{n-1,n}\})$, kde V_i reprezentuje komunitnú sadu. Hrana $E_{i,j}$ medzi objektmi z páru komunitných sád (V_i, V_j) má iný typ, ako hrana $E_{k,l}$ medzi objektmi z páru (V_k, V_l). Matica susednosti (v literatúre označovaná ako bi-adjacency matrix) na podgrafe $(V_i, V_j, E_{i,j})$ zodpovedá heterogénnej relácii $R_{i,j} \subseteq V_i \times V_j$.

Reťaz ovplyvňovateľov $V_1 \xrightarrow{R_{1,2}} V_2 \xrightarrow{R_{2,3}} \ldots \xrightarrow{R_{k-1,k}} V_k$, kde $V_1, \ldots, V_k$ sú komunitné sady a $R_{1,2}, \ldots, R_{k-1,k}$ sú matice susednosti zodpovedajúce reláciám medzi priľahlými komunitnými sadami. Pre každú komunitnú sadu V_i môžeme definovať vektor ohodnotenia r_i, kde platí $|V_i| = |r_i|$. K vektoru r_i vypočítame korešpondujúci vektor r'_{i+1} nasledovne:

$$r'_{i+1} = r_i \cdot R_{i,i+1} \ .$$

Následne vektor r'_{i+1} normalizujeme, čím dostaneme nový vektor r_{i+1}:

$$r_{i+1} = \frac{r'_{i+1}}{max(r'_{i+1})} \ .$$

Pre porovnanie s ostatnými algoritmami ohodnotenia používame štandardnú normalizáciu, kde $\|r_{i+1}\|_2 = 1$.

Vektor r_{i+1} reprezentuje vektor ohodnotenia komunitnej sady V_{i+1}. Tento výpočet opakujeme pre všetky komunitné sady v reťazi ovplyvňovateľov. Posledný vektor ohodnotenia následne použijeme pri výpočte ďalšej iterácie.

Algoritmus 3: Chain of Influencers

Input: set of relationships $R_{1,2}, \ldots, R_{k-1,k}, R_{k,1}$
Output: set of ranked vectors r_i for pool V_i
θ = difference threshold
$r_1 = (1, \ldots, 1)$
while $|r_i - r_i^o ld| > \theta$ **do**
  for $i \leftarrow 1$ **to** k **do**
    $r_{i+1} = r_i \cdot R_{i,i+1}$
    $r_{i+1} = \text{Normalize}(r_{i+1})$
    if *(i+1) mod k = 1* **then**
      $r_1 = r_{i+1}$
    end
  end
end
return r

Zhlukovacie algoritmy s využitím ohodnotenia

Algoritmy PageRank a HITS popísané v kapitolách 3.1 a 3.2 boli vyvinuté na analýzu sietí webových stránok, ktoré sú reprezentované neohodnoteným orientovaným grafom (váhy všetkých hrán = 1). Algoritmus Chain of Influencers popísaný v kapitole 3.3 bol vyvinutý na ohodnotenie heterogénnych sietí a umožňuje vytvárať sémanticky odlišné ohodnotenie všetkých komunitných sád.

Zhlukovanie poskytuje iný pohľad na informačnú sieť a umožňuje nám nájsť štrukúry, ktoré rozdeľujú objekty informačnej siete do podmnožín (zhlukov). Objekty zhluku zdieľajú niektoré spoločné črty. Vzdialenosť medzi objektmi je použitá za účelom zoskupenia podobných objektov do zhluku, pričom oddeľuje odlišné objekty ďaleko od seba. Spektrálne zhlukovanie je jedna z metód, ktoré poskytujú vysokú kvalitu zhlukovania pre homogénne informačné siete a pracuje s maticou susednosti. Pre heterogénne informačné siete môžeme použiť štandardné metódy zhlukovania, výsledky však majú veľmi malú výpovednú hodnotu. Tento problém riešia metódy dolovania podobnosti, napríklad SimRank [22] alebo LinkClus [23] - obe sú však časovo veľmi náročné pre väčšie grafy, pretože počítajú vzdialenosť pre každý pár uzlov v grafe. Spojením ohodnotenia a zhlukovania vznikol algoritmus RankClus.

3.4 RankClus

Algoritmus RankClus [24] výrazne redukuje časovú náročnosť tým, že používa podmienené ohodnotenie na vytvorenie zhlukov a následne počíta vzdialenosť medzi každým objektom a stredom zhluku. To poskytuje vyššiu kvalitu ohodnotenia jednotlivých objektov na základe zhluku, do ktorého patria. Zároveň sa zvyšuje aj kvalita zhlukovania, pretože obidve operácie sú prepojené a navzájom sa ovplyvňujú.

Existuje mnoho druhov heterogénnych informačných sietí. Často používaná a populárna je dvojtypová informačná sieť. Môžeme ňou reprezentovať napríklad vzťah *konferencia - autor* v bibliografickej databáze alebo *film - herec* vo filmovej databáze. Algoritmus RankClus pracuje práve s dvojtypovou informačnou sieťou [24].

Definícia 12 (Dvojtypová informačná sieť): Pre 2 množiny typov objektov X a Y, kde $X = \{x_1, x_2, x_3, \ldots, x_m\}$, $Y = \{y_1, y_2, y_3, \ldots, y_n\}$ existuje graf G = $\langle V, E\rangle$ reprezentujúci dvojtypovú informačnú sieť na typoch X a Y, ak platí $V(G) = X \cup Y$ a $E(G) = \langle o_i, o_j\rangle$, kde $o_i, o_j \in X \cup Y$.

Matica susednosti prepojení $W_{(m+n)\times(m+n)} = \{w_{o_i o_j}\}$, kde sa $w_{o_i o_j}$ rovná váhe prepojenia $\langle o_i, o_j\rangle$ a $G = \langle\{X \cup Y\}, W\rangle$, reprezentuje dvojtypovú informačnú sieť. X a Y používame na označenie množiny objektov a ich typu. Pre jednoduchšiu prácu rozdelíme maticu prepojení na 4 menšie matice: $W_{XX}, W_{XY}, W_{YX}, W_{YY}$, pričom každá menšia matica označuje podsieť typov objektov. Zapíšeme ich nasledovne:

$$W = \begin{pmatrix} W_{XX} & W_{XY} \\ W_{YX} & W_{YY} \end{pmatrix} \ .$$

Definícia 13 (Funkcia ohodnotenia): Pre danú dvojtypovú informačnú sieť $G = \langle\{X \cup Y\}, W\rangle$ existuje funkcia $f : G \to (\vec{r}_X, \vec{r}_Y)$, ktorá ohodnotí každý objekt v type X a Y, kde platí:

$$\forall x \in X, \vec{r}_X(x) \geq 0, \sum_{x \in X} \vec{r}_X(x) = 1 \ ,$$

$$\forall y \in Y, \vec{r}_Y(y) \geq 0, \sum_{y \in Y} \vec{r}_Y(y) = 1 \ .$$

Definícia 14 (Podmienené ohodnotenia a ohodnotenie v rámci zhluku): Pre daný typ X, zhluk $X' \subseteq X$ a podsieť $G' = \langle\{X' \cup Y\}, W'\rangle$ je definovaný graf G indukovaný podmnožinou vrcholov $X' \cup Y$. Podmienené ohodnotenie nad Y, značené ako $\vec{r}_{Y|X'}$ a ohodnotenie v rámci zhluku nad X', značené ako $\vec{r}_{X'|X'}$, sú definované funkciou ohodnotenia f na podsieti $G' : (\vec{r}_{X'|X'}, \vec{r}_{Y|X'}) = f(G')$. Podmienené ohodnotenie nad X, značené $\vec{r}_{X|X'}$, je definované ako ohodnotenie propagácie $\vec{r}_{Y|X'}$ nad sieťou G:

$$\vec{r}_{X|X'}(x) = \frac{\sum_{j=1}^{n} W_{XY}(x, j)\vec{r}_{Y|X'}(j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} W_{XY}(i, j)\vec{r}_{Y|X'}(j)} \ .$$

Podľa tejto definície sú podmienené ohodnotenie nad Y a ohodnotenie v rámci zhluku X' priamočiare, keď sú aplikované ako funkcia ohodnotenia na podsieť G' indukovanú zhlukom X'. Podmienené ohodnotenie nad celou množinou objektov z X je komplexnejšie, pretože nie každý objekt patrí do podsiete G'. Pre daný zhluk X' vypočítame podmienené ohodnotenie nad Y ako $\vec{r}_{Y|X'}$, vďaka čomu môžeme vypočítať podmienené ohodnotenie nad X relatívne k zhluku X' podľa aktuálneho ohodnotenia Y.

Tieto definície môžeme zhrnúť nasledovne: pre dvojtypovú informačnú sieť $G = \langle\{X \cup Y\}, W\rangle$, cieľový typ X a počet zhlukov K, generujeme K zhlukov $\{X_k\}$ na X, na ktoré aplikujeme ohodnotenie v rámci zhluku pre typ X a podmienené ohodnotenie pre typ Y: $\vec{r}_{X|X_k}$, $\vec{r}_{Y|X_k}$ a $k = 1, 2, \ldots, K$.

Jednoduché ohodnotenie: Pre danú informačnú sieť $G = \langle\{X \cup Y\}, W\rangle$ vypočítame jednoduché ohodnotenie typov X a Y nasledovne:

$$\vec{r}_X(x) = \frac{\sum_{j=1}^{n} W_{XY}(x,j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} W_{XY}(i,j)},$$

$$\vec{r}_Y(y) = \frac{\sum_{i=1}^{n} W_{XY}(i,y)}{\sum_{i=1}^{m}\sum_{j=1}^{n} W_{XY}(i,j)}.$$

Časová náročnosť jednoduchého ohodnotenia je $O(|E|)$, kde sa $|E|$ rovná počtu hrán. Zo zápisu vyplýva, že jednoduché ohodnotenie vyjadruje normalizovanú váhu každého uzlu v grafe, pričom považuje všetky hrany za rovnako dôležité.

Autoritné ohodnotenie: Užitočnejšia funkcia ohodnotenia zohľadňuje autoritu uzlu a podľa nej priradí uzlu vyššie ohodnotenie. Môžeme ho demonštrovať na informačnej sieti IMDb, ktorú bližšie popíšeme v kapitole 4. Definujeme nasledovné pravidlá:

- Pravidlo 1: Vysoko ohodnotení režiséri režírujú veľa filmov s vysoko ohodnotenými hercami.
- Pravidlo 2: Vysoko ohodnotení herci hrajú vo veľa filmoch, ktoré režírujú vysoko ohodnotení režiséri.
- Pravidlo 3: Ohodnotenie režiséra je zvýšené, ak spolupracoval s veľkým množstvom hercov alebo s vysoko ohodnotenými hercami.

Spojením týchto troch pravidiel upravíme výpočet ohodnotenia do nasledovnej formy, kde parameter $\alpha \in [0, 1]$ reprezentuje váhu autoritného ohodnotenia:

$$\vec{r}_Y(i) = \alpha \sum_{j=1}^{m} W_{YX}(i,j)\vec{r}_X(j) + (1-\alpha)\sum_{j=1}^{n} W_{YY}(i,j)\vec{r}_Y(j) \ .$$

Odhad koeficientov pomocou algoritmu EM: Pre zmiešaný model potrebujeme odhadnúť správne hodnoty koeficientov, ktoré reprezentujeme maticou $\Theta_{m\times K} = \{\pi_{i,k}\}(i = 1, 2, \ldots, m; k =$

$1, 2, \ldots, K$). Naším cieľom je odhadnúť najlepšiu Θ podľa hrán v sieti. Využijeme na to algoritmus EM [25] (expectation–maximization).

Výpočet centra zhluku a vzdialenosti: Po odhadnutí koeficientov pre každý objekt x_i zmiešaného modelu môžeme x_i reprezentovať k-dimenzionálnym vektorom $\vec{s}_{x_i} = (\pi_{i,1}, \pi_{i,2}, \ldots, \pi_{i,k})$. Centrum zhluku vypočítame ako strednú hodnotu $\vec{s}_{x_i}$ pre všetky x_i v zhluku, kde $|X_k|$ je veľkosť zhluku k:

$$\vec{s}_{X_k} = \frac{\sum_{x \in X_k} \vec{s}(x)}{|X_k|} \quad .$$

Vzdialenosť medzi objektom x a centrom zhluku X_k definujeme pomocou kosínusovej podobnosti:

$$D(x, X_k) = 1 - \frac{\sum_{l=1}^{K}] \vec{s}_x(l) \vec{s}_{X_k}(l)}{\sqrt{\sum_{l=1}^{K} (\vec{s}_x(l))^2} \sqrt{\sum_{l=1}^{K} (\vec{s}_{X_k}(l))^2}} \quad .$$

Všeobecný princíp fungovania algorimu pozostáva z prevedenia objektov do $\vec{s}_x$ podľa zmiešaného modelu, vytvorenia počiatočných zhlukov a následného priradenia objektu do najbližšieho zhluku X_k. Tento proces sa iteratívne opakuje, kým sa zhluky medzi iteráciami dostatočne líšia. Kvalita zhlukov sa zvyšuje, pretože navzájom podobné objekty sú zoskupené blízko seba. Spolu s kvalitou zhlukov sa zvyšuje aj kvalita ohodnotenia jednotlivých objektov v rámci zhluku. Algoritmus funguje v troch krokoch, ktoré popíšeme bližšie:

- Krok 0: Inicializácia
  V inicializačnom kroku vygenerujeme počiatočné zhluky pre cieľové objekty a priradíme ich do náhodného zhluku $X_{1\ldots K}$

- Krok 1: Ohodnotenie každého zhluku
  Podľa existujúcich zhlukov vypočítame podmienené ohodnotenie pre typy Y,X a ohodnotenie v rámci zhluku pre typ X. V tomto kroku musíme taktiež skontrolovať, či niektorý zo zhlukov nie je prázdny. To môže byť zapríčinené nesprávnou inicializáciou z dôvodu náhodného priradenia objektov do zhluku alebo predpojatými výsledkami algoritmu. Ak takáto situácia nastane, musíme algoritmus spustiť znovu od kroku 0.

- Krok 2: Odhad koeficientov zmiešaného modelu
  Pre zmiešaný model odhadneme parameter Θ použitím algoritmu EM, čím dostaneme novú reprezentáciu objektu $\vec{s}_X$ a stredu zhluku $\vec{s}_{X_k}$. Počet iterácií výpočtu Θ môže byť nízky, najlepšie výsledky [24] dosahoval $t = 5$.

- Krok 3: Úprava zhlukov
  Vypočítame vzdialenosť medzi objektom a všetkými stredmi zhlukov. Objekt následne priradíme do najbližšieho zhluku.

- Kroky 1-3 opakujeme, kým sa zhluky dostatočne líšia (špecifikované parametrom ε), prípadne počtom iterácií c.

Algoritmus formálne popisuje nasledovný pseudokód:

Algoritmus 4: RankClus

Input: Bi-type information network $G = \langle X, Y; W \rangle$, number of clusters k, number of iterations c

Output: k clusters $X_i, \vec{r}_{X_i|X_i}, \vec{r}_{Y|X_i}$

`/* Step 0: Initialization */`

$t = 0$

$\{X_i^{(t)}\}_{i=1}^{K}$ = initial partitions for X

while *difference in clusters* $> \varepsilon$ // *iteration count* $< c$ **do**

  `/* Step 1: Ranking for each cluster */`

  for $i \leftarrow 1$ **to** k **do**

    $G_i^{(t)} = \text{GetSubgraph}(X_i^{(t)}, Y)$

    $\vec{r}_{X_i|X_i}^{(t)}, \vec{r}_{Y|X_i}^{(t)} = f(G_i^{(t)})$

    $\vec{r}_{X_i|X_i}^{(t)} = W_{XY}\vec{r}_{Y|X_i}^{(t)}$

  end

  `/* Step 2: Mixture model coefficient estimation */`

  Evaluate Θ for each x_i

  for $i \leftarrow 1$ **to** k **do**

    $\vec{s}_{X_k}^{(t)} = \text{GetCenters}(X_k^{(t)})$

  end

  `/* Step 3: Cluster adjustement */`

  for $x \in X$ **do**

    for $i \leftarrow 1$ **to** k **do**

      Calculate Distance$(x, X_k^{(t)})$

    end

    Assign x to $X_{k_0}^{t+1}$, where $k_0 = argmin_k D(x, X_k^t)$

  end

end

4 Databáza IMDb

IMDb[7] je online databáza filmov, televíznych seriálov, domácich videí a videohier. Obsahuje informácie o jednotlivých dielach, ich hercoch, režiséroch, zhrnutia zápletiek, užívateľské recenzie a hodnotenia. K decembru 2018 stránka obsahuje približne 5,3 miliónov titulov, 9,3 miliónov osobností a 83 miliónov registrovaných užívateľov.

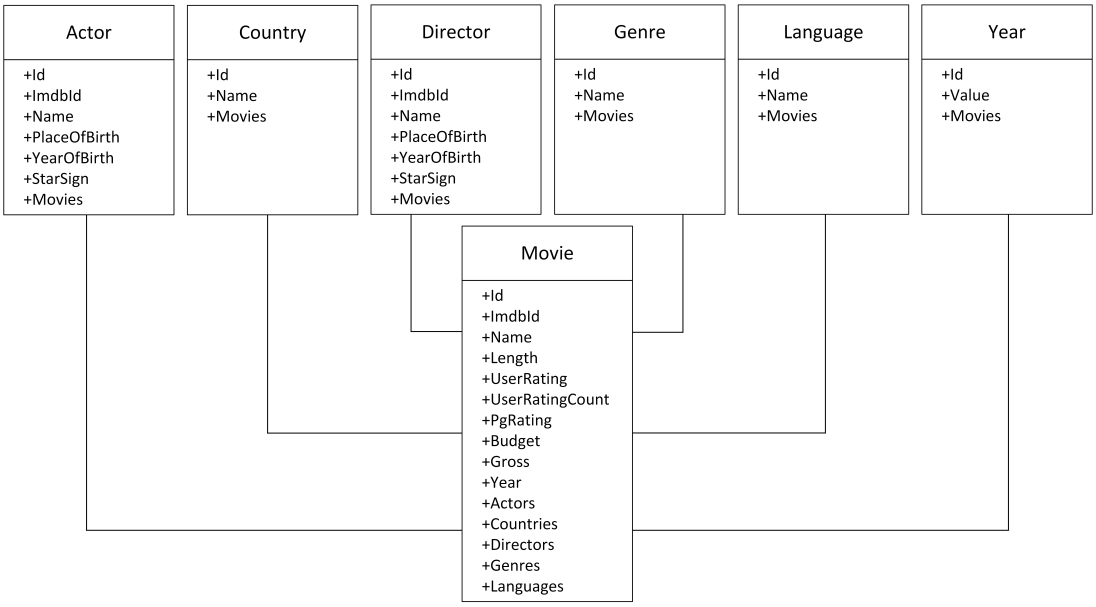
IMDb založil 17. októbra 1990 filmový fanúšik a programátor Col Needham. Prvotný obsah získal z iných, menších webov pomocou unixových shell skriptov. Do konca roku 1990 webstránka obsahovala informácie o približne 10000 filmoch. V roku 1996 vznikla spoločnosť Internet Movie Databaze Ltd., pod ktorú stránka oficiálne patrila. Neskôr bola v roku 1998 odkúpená spoločnostou Amazon.com, Inc. a zaradená ako dcérska spoločnosť.

Väčšina dát v databáze je pridávaná dobrovoľnými užívateľmi. Webstránka umožňuje registrovaným užívateľom pridávať nové materiály alebo upravovať existujúce. Užívatelia s dobrou reputáciou môžu pridávať a upravovať obsah okamžite, noví uživatelia musia čakať na schválenie obsahu komunitou - do určitej miery je tak zaručená presnosť dát v databáze. Registrovaní užívatelia môžu hodnotiť akékoľvek dielo na škále 1 až 10 bodov, celkové hodnotenie sa následne počíta ako vážený priemer. Databáza ponúka široké množstvo filtrov, na základe ktorých je možné vyhľadať jednotlivé diela alebo podmnožiny diel.

IMDb neposkytuje verejné API pre automatizovaný prístup k dátam, preto sme databázu pre potreby diplomovej práce museli získať technikou *scrapovania*[8]. Scrapovať všetky diela je nepraktické, rozhodli sme sa preto vyfiltrovať podmnožinu filmov od roku 1985 až po rok 2017. Najskôr sme získali zoznam filmov spadajúcich do nami špecifikovaného rozsahu, následne sme ich uložili do dočasnej databázy spolu s ich metadátami (unikátny identifikátor filmu, názov, rok produkcie a prípadnu chybu pri získavani dát). Následne prebiehalo scrapovanie jednotlivých filmov, pre ktoré sme ukladali povinné parametre (herci, režiséri) ako aj nepovinné parametre (žánre, do ktorých film patrí, užívateľské hodnotenie, počet užívateľských hodnotení, MPAA hodnotenie, rozpočet a zisk). Na spracovanie HTML obsahu sme použili knižnicu HtmlAgilityPack[9]. Získavanie dát bežalo paralelne na 16 vláknach, zabralo približne 42 hodín a po sieti sa prenieslo zhruba 480GB dát. Štruktúra dát určovala databázové modely, ktoré sú zachytené v obrázku 7.

Dáta sme ukladali do databázy MySQL[10], ktorá bola zvolená z dôvodu relatívne dobrého výkonu pri paralelných prístupoch k dátam (oproti napríklad SQLite). Informácie o filmoch boli ukladané v transakcii, aby bola zaručená konzistencia dát. Pri scrapovaní sme taktiež kontrolovali duplicitné záznamy podľa unikátneho IMDb identifikátora, čím sme výrazne zredukovali množstvo sťahovaných dát. Počas scrapovania sme ukladali informácie o osobách do samostatnej tabuľky, na ktorú boli naviazaní herci a režiséri. Neskôr sme však kvôli zvýšenej zložitosti

[7] https://www.imdb.com/
[8] web scraping - technika dolovania dát z HTML obsahu
[9] https://html-agility-pack.net/
[10] https://www.mysql.com/



Obr. 7: UML diagram databázového modelu

generovaných SQL dopytov na databázu túto tabuľku odstránili a osobné údaje sme priradili priamo do tabuliek hercov a režisérov.

Na prístup k databáze sme zvolili Entity Framework Core[11], ktorý nám umožňuje jednoduchú prácu so záznamami. Pomocou ORM poskytuje spôsob, akým tvoriť komplexné SQL dopyty, ktoré budeme neskôr využívať na filtrovanie načítavaných dát.

Výsledná databáza sa skladá z nasledovných typov objektov: film, režisér, herec, rok, krajina a jazyk. Všetky objekty sú navzájom prepojené cez film a tvoria hviezdicovú štruktúru.

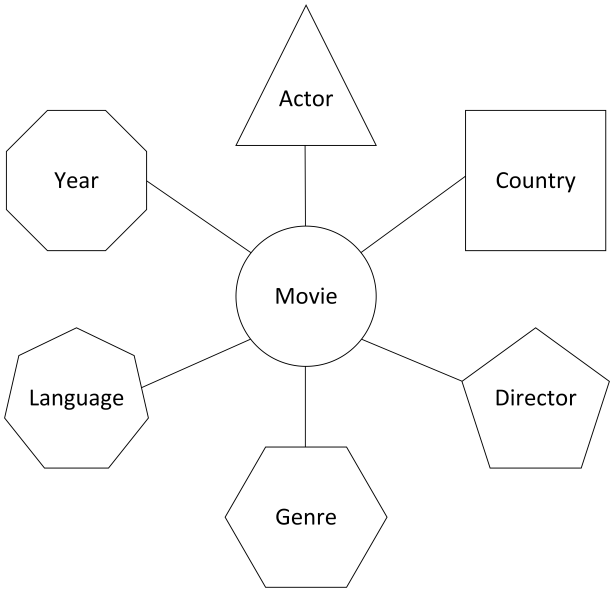
Niektoré záznamy obsahovali chýbajúce povinné atribúty. Rozhodli sme sa preto tieto záznamy odstrániť, čím sme dostali nasledujúce počty záznamov v tabuľkách databázy, zobrazené v tabuľke 2.

Tabuľka 2: Počet záznamov v tabuľkách databázy

Názov tabuľky	Počet záznamov
Herci	1 485 797
Filmy	191 979
Režiséri	92 452
Krajiny	757
Jazyky	288
Roky	33
Žánre	27

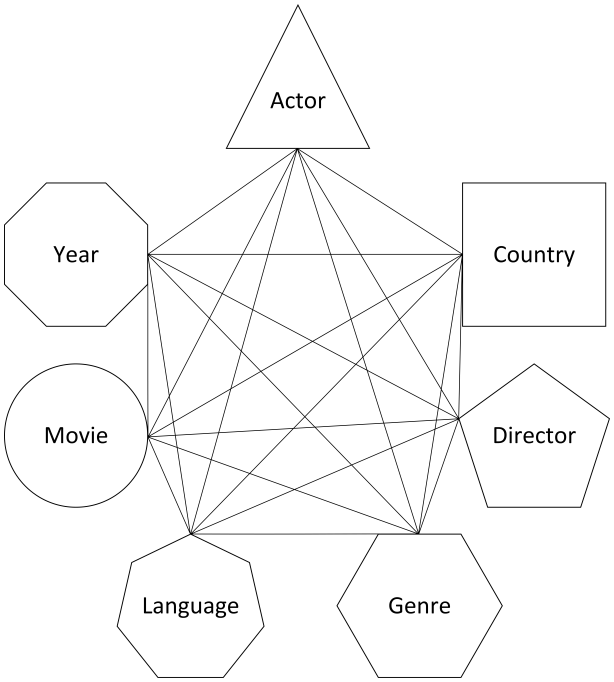
Z tabuľky 2 vidíme, že medzi filmom a ostatnými objektmi nie je relácia 1:1. Vo filme hrá niekoľko hercov, ktorí môžu hrať vo viacerých filmoch. To isté platí aj o ostatných reláciách,

[11]https://docs.microsoft.com/en-us/ef/core/



Obr. 8: Hviezdicová schéma databázy

preto sú väzby medzi tabuľkou filmov a ostatnými tabuľkami databázy typu M:N. Pre potreby analýzy rozšírime schému siete z hviezdicovej na úplný graf. Doplnením väzobných tabuliek medzi všetky dvojice vzniká multipartitná sieť.



Obr. 9: Rozšírená schéma databázy

5 Implementácia

Pri implementácii sme sa rozhodli použiť moderné technológie, ktoré nám umožnia efektívne spracovať širokú škálu dátových sád. Taktiež nám umožnia jednoduché rozšírenie programu o ďalšiu funkcionalitu.

.NET Core Jadro programu používa open-source framework .NET Core, vďaka ktorému môžeme program spustiť podľa potreby na ľubovoľnej platforme. Vytvorené užívateľské rozhranie využíva grafický podsystém WPF.

Entity Framework Core Na prácu s databázou bol vybratý ORM Entity Framework Core, pomocou ktorého môžeme jednoducho modelovať databázu definovanú modelmi v kóde. Generovaním SQL dopytov výrazne uľahčuje prácu s databázou. Podporuje veľké množstvo relačných databázových systémov.

MySQL Dátovú sadu IMDb sme ukladali do populárnej relačnej databázy MySQL.

5.1 Projektová štruktúra

Výsledný program bol počas vývoja rozdelený do niekoľkých modulov, ktoré sú navzájom previazané. Tabuľka 3 zobrazuje popis jednolivých modulov.

Tabuľka 3: Popis modulov programu

Modul	Popis
Database	Databázová vrstva
Gui	Užívateľské rozhranie
Models	Zdieľané modely pre ostatné moduly programu
Networks	Implementácia algoritmov a spracovanie grafov
Scraper	Modul na získanie databázy IMDb
Services	Zdieľané služby pre ostatné moduly programu

5.2 Rozšírenie databázovej schémy

Pôvodnú schému databázy sme rozšírili o väzobné tabuľky, čím vznikol úplný graf. Tým odpadla nutnosť spájať jednotlivé tabuľky cez ich väzbu na film. Pôvodný SQL dopyt (Výpis 1) sa tak zjednoduší na (Výpis 2) bez nutnosti spájať tabuľky a následne odstraňovať duplicity. Dôjde tým aj k zrýchleniu načítavania dát, keď pri testovaní SQL dopyt (Výpis 1) trval približne 1,47 sekundy a SQL dopyt (Výpis 2) trval približne 0,37 sekundy. Obidva dopyty používali podmnožinu 19984 režisérov.

```
SELECT DISTINCT DM.DirectorId, GM.GenreId
FROM DirectorMovie AS DM JOIN genremovie AS GM ON DM.MovieId = GM.MovieId
WHERE DM.DirectorId IN (...)
```

Výpis 1: SQL dopyt na získanie vzťahu medzi režisérom a žánrom cez film

```
SELECT DISTINCT DirectorId, GenreId
FROM DirectorGenre
WHERE DM.DirectorId IN (...)
```

Výpis 2: SQL dopyt na získanie vzťahu medzi režisérom a žánrom z väzobnej tabuľky

5.3 Vytváranie SQL dopytov

Generovanie SQL dopytov musí prebiehať automaticky, podľa užívateľom špecifikovaného filtra. Entity Framework nám umožňuje definovať lambda výraz, ktorý následne vložíme do metódy `DbSet<T>.Where()`, čím vyfiltrujeme záznamy podľa nami vybraných atribútov a ich hodnôt. Pri kompilácii však presne nevieme, s akým typom objektov budeme pracovať a následne filtrovať. Všetky databázove modely implementujú interface `IDbModel` a väzobné tabuľky `IDbManyToManyModel`. To nám následne umožní použiť reflexiu a pomocou nej ponúknuť užívateľovi atribúty modelu na filtrovanie. Použijeme ju tiež na vytvorenie vstupu do generátora lambda výrazov.

Generovanie SQL dopytov pomocou reflexie umožňuje jednoduchú zmenu databázovych modelov, prípadne celej databázy. Výsledný program a jeho algoritmy teda môžeme aplikovať na širokú škálu dátových sád.

Podľa dátového typu atribútu ponúkneme užívateľovi jeden z dvoch filtrov: `EqualsFilter` alebo `ConstrainedFilter`. `EqualsFilter` sa používa primárne na textové atribúty, ale možeme ho aplikovať aj na číselné hodnoty v prípade, že chceme filtrovať konkrétnu hodnotu atribútu. `ConstrainedFilter` zase poskytuje možnosť filtrovať inkluzívny rozsah hodnôt. Filtre môžeme navzájom kombinovať, a to nám následne umožní vybrať z databázy ľubovoľnú podmnožinu záznamov.

Doménové modely používané v programe sú vyhľadávané metódou `GetModelTypes` (Výpis 3) v triede `ModelReverser`. Tá najskôr pomocou reflexie vyhľadá všetky typy modelov v mennom priestore aplikácie a následne vyberie tie, ktoré implementujú rozhranie `IDbModel`. Každý typ modelu spracujeme metódou `GetModelProperties`, čo nám umožní jednoduchšiu prácu pri zobrazení v užívateľskom rozhraní aj pri vytváraní filtrov.

```
public static List<Type> GetModelTypes()
{
   const string namespaceToSearch = "Imdb.Database.Model";
   return AppDomain.CurrentDomain.GetAssemblies()
      .SelectMany(t => t.GetTypes())
      .Where(t => t.Namespace == namespaceToSearch && t.IsClass && t.
         GetInterfaces()
      .Contains(typeof(IDbModel))).ToList();
}
```

Výpis 3: Vyhľadanie typov modelov

S pripravenými modelmi môžeme generovať SQL dopyt. Jazyk C# neumožňuje špecifikovať konkrétny dátový typ generickej metódy z jeho názvu počas kompilácie programu, preto sme aj tento problém riešili cez reflexiu. Najskôr voláme negenerickú metódu (Výpis 4), v ktorej vytvoríme inštanciu konkrétneho dátového typu, a ten následne reflexiou vložíme do generickej metódy (Výpis 5).

```
public static IQueryable<IDbModel> Query(string modelName, IEnumerable<IFilter>
    filters = null)
{
   var instance = CreateInstanceOf(modelName);
   return (IQueryable<IDbModel>) typeof(QueryGenerator)
      .GetMethod("QueryGeneric")
      .MakeGenericMethod(instance.GetType())
      .Invoke(null, new object[] {filters});
}
```

Výpis 4: Negenerická metóda generovania SQL dopytu

V generickej metóde sa už zostavuje samotný lambda výraz pre konkrétny dátový typ. Výsledný lambda výraz vložíme do metódy `Where` databázového kontextu, Entity Framework nám z neho vygeneruje SQL dopyt a následne vráti požadované dáta.

```
public static IQueryable<T> QueryGeneric<T>(IEnumerable<IFilter> filters) where
    T : class, IDbModel
{
   var parameter = Expression.Parameter(typeof(T));
   var expressionList = new List<BinaryExpression>();
```

```
    if (filters != null) {/* Create filter expressions and insert into list */}
    Expression body = null;
    for (var expressionIndex = 0; expressionIndex < expressionList.Count;
        expressionIndex++) {/* Create expression body */}
    if (body != null)
    {
        var lambda = Expression.Lambda<Func<T, bool>>(body, parameter);
        return Context.Set<T>().Where(lambda);
    }

    return Context.Set<T>();
}
```

Výpis 5: Generická metóda generovania SQL dopytu

Generická metóda (Výpis 5) zobrazuje len krátky náhľad na funkcionalitu, úplnú implementáciu nájdeme v priloženom zdrojovom kóde. Najprv v cykle iterujeme filtre, z ktorých vytvárame výrazy obsahujúce vymedzenia špecifikované filtrom a tie vkladáme do listu. Následne tieto výrazy iterujeme a skladáme z nich telo lambda výrazu. V prípade, že užívateľ nešpecifikoval žiadne filtre, vrátime celú množinu dát.

5.4 Komunitné sady

Po načítaní bázovej sady pokračujeme vytvorením a načítaním ostatných relácií spojených s bázovou sadou. Relácie priamo spojené s bázovou sadou sa vytvárajú metódou `Create` triedy `RelationshipBuilder`, ktorá očakáva 3 vstupné parametre: `partityAName`, `partityBName` a `partityAIds`. Prvé dva parametre obsahujú názov partít, medzi ktorými má vzniknúť relácia. Posledný parameter je zoznam unikátnych identifikátorov, podľa ktorých filtrujeme. Pri generovaní SQL dopytov predpokladáme lexikografické usporiadanie názvov databázových modelov, preto musíme prvé dva parametre správne zoradiť. Pri vytváraní relácie herca a filmu tak vznikne SQL dopyt na tabuľku *ActorMovie*. `RelationshipBuilder.Create` interne volá metódu `QueryGenerator.ForAinAIds`, v ktorej cez reflexiu vytvoríme inštanciu dátového modelu väzobnej tabuľky pre nami špecifikovanú reláciu. Tá následne volá metódu `ForAinAIdsGeneric` (Výpis 6) triedy `QueryGenerator`, ktorá podľa identifikátorov vygeneruje lambda výraz, vytvorí SQL dopyt a následne načíta záznamy z databázy.

```
public static List<(int, int)> ForAinAIdsGeneric<T>(string partityA, string
    partityB, IEnumerable<int> partityAIds) where T : class, IDbManyToManyModel
{
    string selectAId = $"{partityA}Id";
```

```
    string selectBId = $"{partityB}Id";
    var parameter = Expression.Parameter(typeof(T));
    var method = partityAIds.GetType().GetMethod("Contains");
    var call = Expression.Call(Expression.Constant(partityAIds), method,
        Expression.Property(parameter, selectAId));
    var whereLambda = Expression.Lambda<Func<T, bool>>(call, parameter);
    var selectProp = Expression.New(
       typeof((int, int)).GetConstructor(new[] {typeof(int), typeof(int)}),
       Expression.Property(parameter, selectAId),
       Expression.Property(parameter, selectBId));
    var selectLambda = Expression.Lambda<Func<T, (int, int)>>(selectProp,
        parameter);

    return Context.Set<T>().Where(whereLambda).Select(selectLambda).ToList();
}
```

Výpis 6: Generická metóda generovania bázovej sady

Ako príklad môžeme uviesť bázovú sadu hercov, ku ktorým načítame ostatné priamo prepojené relácie. Pre meta cestu *AMG* voláme metódu `Create`, do ktorej vložíme príslušné parametre - v tomto prípade `Create("Actor", "Movie", new [] {18,42,59})`. Pri volaní tejto metódy vznikne SQL dopyt (Výpis 7). Prvé dva parametre určujú partity hercov a filmov, medzi ktorými vznikne relácia. Posledný parameter je zoznam unikátnych identifikátorov prvej partity - hercov - podľa ktorých budeme filtrovať databázové záznamy. Na zachytenie a reprezentáciu vzťahov medzi partitami používame nami definovaný dátový typ `Relationship`. Do tohto dátového typu ukladáme názvy partít relácie, zoznamy unikátnych databázových identifikátorov a názov objektu pre každý unikátny identifikátor. Podobným spôsobom získame z databázy aj ostatné potrebné údaje.

```
SELECT ActorId, MovieId
FROM ActorMovie
WHERE ActorId IN (18,42,59)
```

Výpis 7: SQL dopyt metódy ForAinAIds

Pre zvyšok relácií už poznáme zoznam unikátnych identifikátorov a načítavame záznamy, ktoré sú previazané s bázovou sadou. Využívame na to metódu `RelationshipBuilder.Join`, ktorá interne volá metódu `QueryGenerator.BetweenAandB`. Pomocou reflexie opätovne vytvoríme inštanciu modelu väzobnej tabuľky, ktorú vložíme do metódy `BetweenAandBGeneric` triedy `QueryGenerator`. Výsledkom je SQL dopyt (Výpis 8), ktorého vrátené záznamy taktiež uložíme

do dátového typu `Relationship`. Tento postup opakujeme pre ostatné relácie. Výsledkom je 21 objektov typu `Relationship`, ktoré predstavuje komunitné sady.

```
SELECT GenreId, MovieId
FROM GenreMovie
WHERE GenreId IN (4,19,65,8) AND MovieId IN (11,32,38,74)
```

Výpis 8: SQL dopyt metódy BetweenAandB

5.5 Meta cesty

Po načítaní komunitných sád môžeme pokračovať vytváraním meta ciest. Po definovaní meta cesty vytvoríme nový zoznam relácií, ktorý zodpovedá vybraným dátovým typom meta cesty. Pri vytváraní tohto zoznamu tiež kontrolujeme, či vytvorená meta cesta reprezentuje homogénnu alebo heterogénnu sieť.

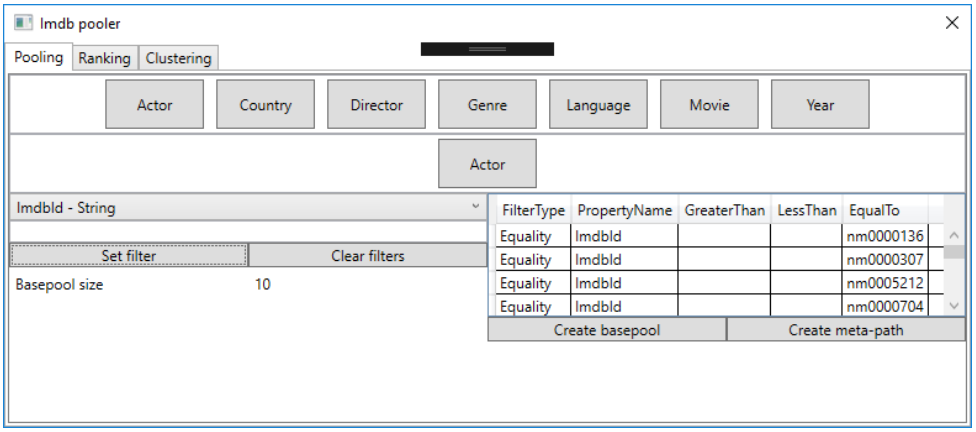
Po vytvorení zoznamu relácií vygenerujeme váženú maticu susednosti. Ako príklad môžeme uviesť meta cestu *AMDMA*, pre ktorú vytvoríme maticu susednosti nasledovne:

$$\begin{aligned} R_{AA}^{MDM} &= R_{AM} \cdot R_{MD} \cdot R_{DM} \cdot R_{MA} = \\ &= R_{AM} \cdot R_{MD} \cdot R_{MD}^{T} \cdot R_{AM}^{T} = \\ &= (R_{A}M \cdot R_{MD}) \cdot (R_{AM} \cdot R_{MD})^{T} = \\ &= R_{AD}^{M} \cdot (R_{AD}^{M})^{T} \end{aligned}$$

5.6 Užívateľské rozhranie

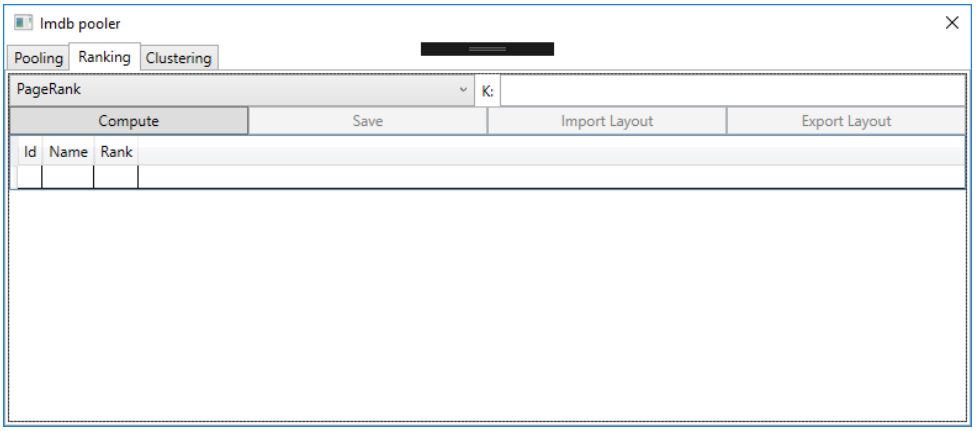
Užívateľské rozhranie programu je logicky rozdelené na zobrazenia, pomocou ktorých špecifikujeme vstup pre jednotlivé algoritmy. Hlavné zobrazenie (Obrázok 10) nám umožňuje definovať bázovú sadu z dostupných databázových modelov. Pre každý model sú načítané jeho atribúty, podľa ktorých môžeme špecifikovať filter bázovej sady. V obrázku 10 sme definovali bázovú sadu 10 hercov pomocou ich unikátneho IMDb identifikátora. Následným kliknutím na tlačidlo `Create basepool` načítame záznamy bázovej sady a priľahlých komunitných sád z databázy. Po načítaní záznamov pokračujeme vytvorením meta cesty. Tú rovnako definujeme z dostupných databázových modelov. Zostavenie relácií, grafov a výslednej matice susednosti vyvoláme kliknutím na tlačidlo `Create meta-path`. Následne môžeme pokračovať spúšťaním algoritmov.

V zobrazení ohodnotenia (Obrázok 11) máme na výber algoritmy popísané v kapitole 3. Po výbere algoritmu ohodnotenia nastavíme parameter K v prípade, že ho algoritmus očakáva ako jeden zo svojich vstupov. Kliknutím na tlačidlo `Compute` spustíme vybraný algoritmus.



Obr. 10: Hlavné zobrazenie užívateľského rozhrania

Po úspešnom vykonaní algoritmu je nám umožnené uloženie výsledkov do formátu `GEXF`[12] pre algoritmy PageRank a HITS, prípadne `TSV` pre Chain of Influencers a RankClus. Taktiež môžeme importovať a exportovať rozmiestnenie vrcholov grafu pre formát `GEFX`. Na vizualizáciu grafov následne použijeme nástroj Gephi[13].



Obr. 11: Zobrazenie ohodnotenia užívateľského rozhrania

[12]https://gephi.org/gexf/format/
[13]https://gephi.org/

6 Experimenty

V tejto kapitole popíšeme vybrané experimenty, ktoré sme vykonali nad projekciami heterogénnych sietí na homogénne siete pre algoritmy PageRank, HITS. Algoritmus Chain of Influencers pracuje priamo s heterogénnymi reláciami. Pre heterogénne siete reprezentované bipartitným grafom sme na experimenty použili algoritmus RankClus. Bázová sada zvolená pre experimenty bola zámerne limitovaná veľkosťou z dôvodu prehľadnejšej vizualizácie grafom a tabuľkou.

Do experimentov neboli zahrnuté symetrické meta cesty, ktorých výsledkom je symetrická matica susednosti. Symetrickú maticu nemá zmysel analyzovať pomocou algoritmov PageRank a HITS, pretože vrátia identické ohodnotenie všetkých vrcholov grafu.

Pre porovnanie sme v experimentoch použili váženú aj binarizovanú maticu. Binarizáciu matice zapíšeme nasledovne:

$$A_{i,j} = \begin{matrix} 0 \Leftrightarrow R_{i,j} = 0 \\ 1 \Leftrightarrow R_{i,j} \neq 0 \end{matrix} .$$

6.1 Bázová sada hercov

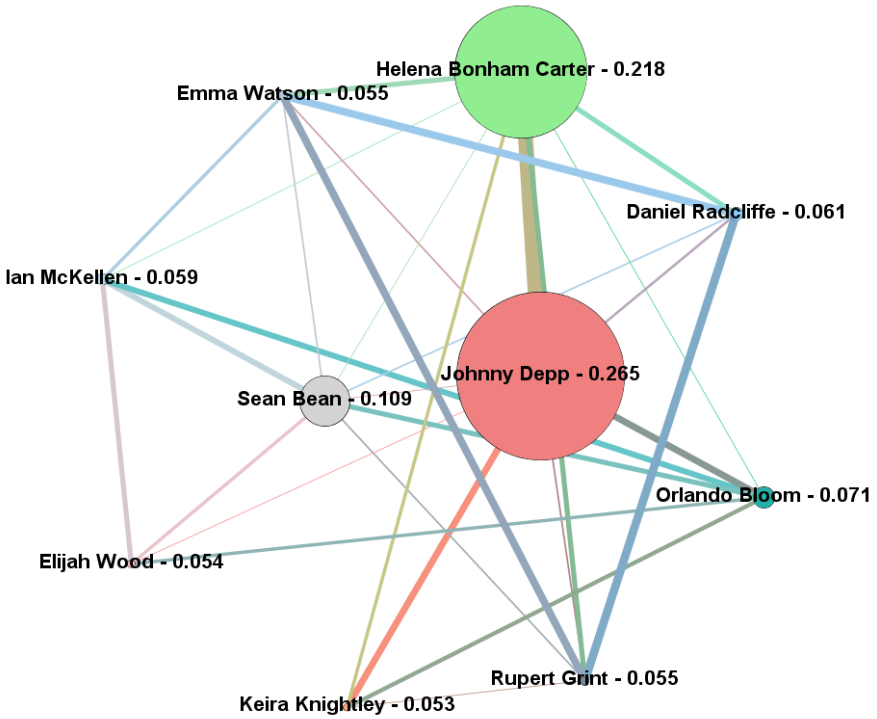
Bázová sada obsahuje spolu 10 hercov:

1. Helena Bonham Carter, Daniel Radcliffe, Rupert Grint a Emma Watson z oktalógie Harry Potter
2. Johnny Depp, Orlando Bloom a Keira Knightley z trilógie Piráti Karibiku
3. Elijah Wood, Ian McKellen a Sean Bean z trilógie Pán Prsteňov

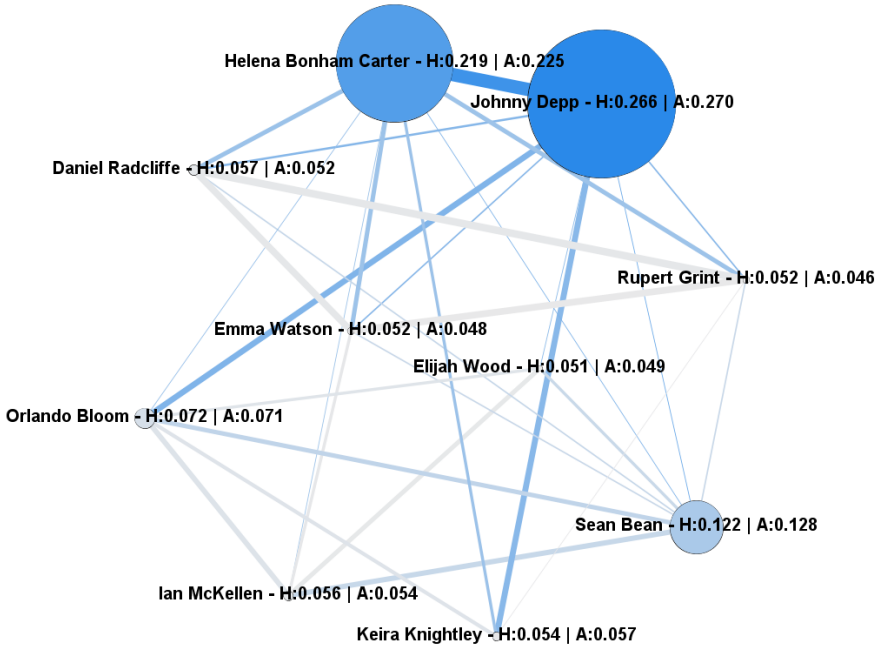
6.2 Meta cesta AMDA

Meta cesta AMDA reprezentuje reláciu hercov R_{AA}^{MD}. Reláciu sme projekciou previedli na homogénnu sieť. Experimenty v rámci tejto meta cesty sme vykonávali pre váženú aj binarizovanú maticu.

Algoritmus PageRank (Obrázok 12) ohodnotil hercov váženej matice na základe počtov filmov, v ktorých hrali, ako aj režisérov, s ktorými spolupracovali. Vysoké ohodnotenie dosiahli herci Johnny Depp a Helena Bonham Carter, ktorí sa spolu často stretli pri natáčaní veľkého množstva filmov. Herec Sean Bean sa taktiež umiestnil relatívne vysoko - hral vo väčšom počte filmov ako dvaja vyššie umiestnení herci. Napriek tomu dosiahol nižšie ohodnotenie. Obdobné výsledky (Obrázok 13) pre váženú maticu dosiahol aj algoritmus HITS.



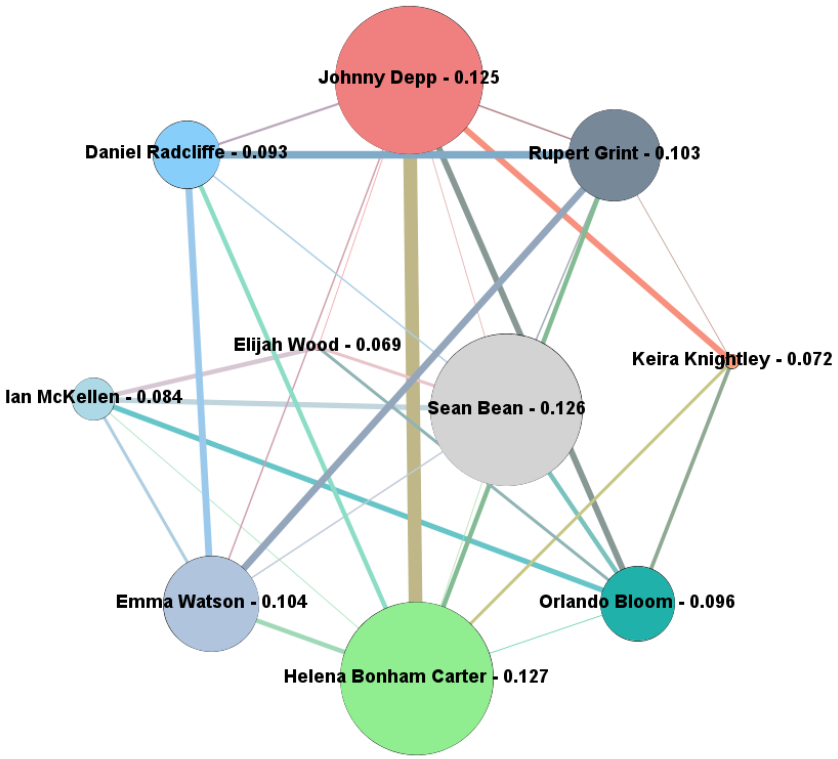
Obr. 12: Meta cesta AMDA, algoritmus PageRank, vážená matica, r_0



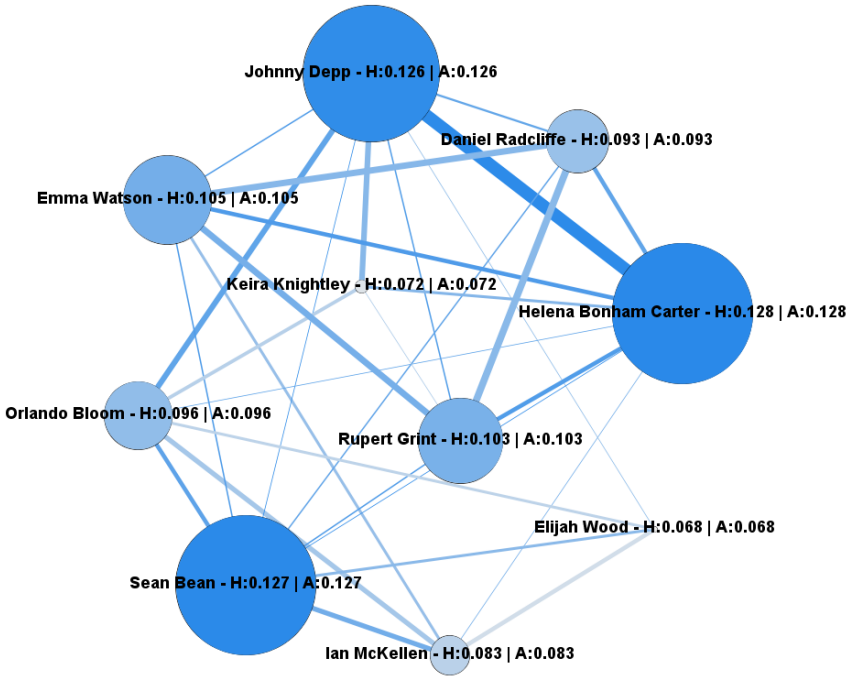
Obr. 13: Meta cesta AMDA, algoritmus HITS (k=10), vážená matica, r_1

Vo vizualizácii výsledkov algoritmu PageRank reprezentuje veľkosť vrcholu jeho ohodnotenie. Vo vizualizácii výsledkov algoritmu HITS potrebujeme zachytiť ohodnotenie autorít aj centier. Preto sme sa rozhodli ohodnotenie autority reprezentovať veľkosťou vrcholu a ohodnotenie centra intenzitou zafarbenia vrcholu grafu. V popise obrázkov a tabuliek tiež uvádzame označenie ohodnotenia, ktoré neskôr používame pri vyhodnotení kvality algoritmov.

Pre binárne matice dopadlo ohodnotenie výrazne odlišne oproti váženej matici. Je to z dôvodu, že zanedbávame počet výskytov hercov vo filmoch a s koľkými režisérmi spolupracovali. Takúto sieť ohodnotíme len na základe existencie prepojenia medzi hercami. Algoritmy PageRank (Obrázok 14) a HITS (Obrázok 15) dosiahli na binárnej matici porovnateľné výsledky.



Obr. 14: Meta cesta AMDA, algoritmus PageRank, binárna matica, r_2



Obr. 15: Meta cesta AMDA, algoritmus HITS (k=10), binárna matica, r_3

Posledný algoritmus aplikovaný na meta cestu je Chain of Influencers (Tabuľka 4), ktorý hodnotí jednotlivé relácie v sieti. Vysoké ohodnotenie hercov korešponduje s ohodnotením váženej matice algoritmami PageRank a HITS. Ohodnotenie filmov a režisérov dopadlo podľa očakávaní, najvyššie ohodnotenie získal režisér Tim Burton, ktorý spolupracoval na množstve filmov s hercami Johnny Depp a Helena Bonham Carter. Vysoko ohodnotené filmy pochádzajú z jeho tvorby. Celkové nízke ohodnotenie filmov a režisérov je spôsobené ich veľkým počtom v bázovej sade.

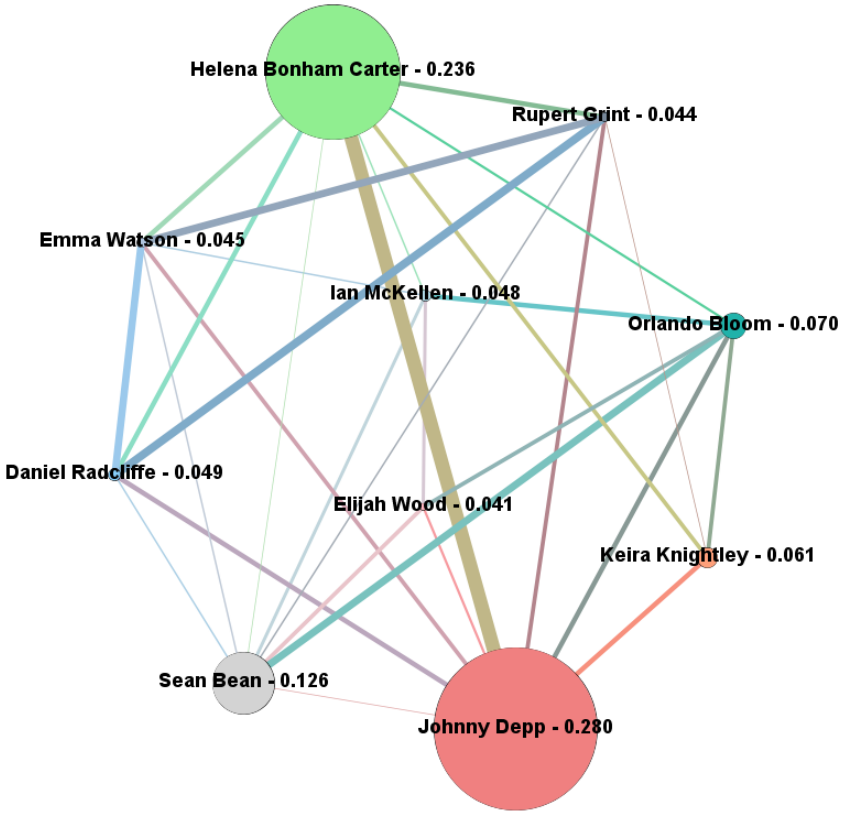
Tabuľka 4: Meta cesta AMDA, algoritmus Chain of Influencers, r_4

Actor	r_A	Movie	r_M	Director	r_D
Johnny Depp	0.28	Corpse Bride	0.01	Tim Burton	0.06
Helena B. Carter	0.24	Charlie and the Chocolate Fa...	0.01	Tom Clegg	0.03
Sean Bean	0.13	Sweeney Todd: The Demon B...	0.01	Gore Verbinski	0.03
Orlando Bloom	0.07	Alice in Wonderland	0.01	David Yates	0.03
Keira Knightley	0.06	Dark Shadows	0.01	Peter Jackson	0.02
Daniel Radcliffe	0.05	The Lone Ranger	0.01	Kenneth Branagh	0.01
Ian McKellen	0.05	Alice Through the Looking G...	0.01	James Ivory	0.01
Emma Watson	0.04	Pirates of the Caribbean: The...	0.01	Mike Newell	0.01
Rupert Grint	0.04	Pirates of the Caribbean: Dea...	0.01	Terry Gilliam	0.01
Elijah Wood	0.04	Pirates of the Caribbean: At...	0.01	Lasse Hallström	0.01

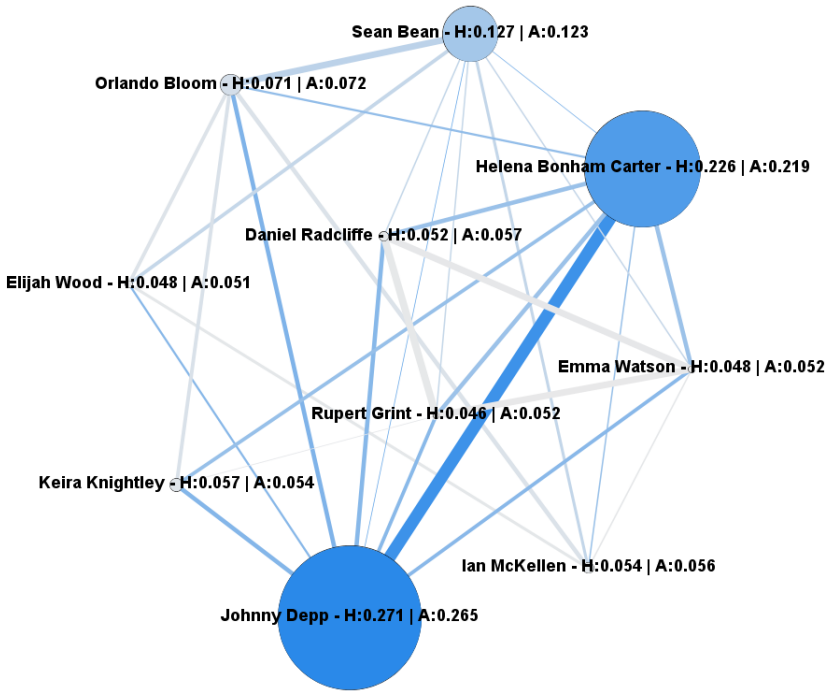
6.3 Meta cesta ADMA

Meta cesta ADMA reprezentuje reláciu hercov R_{AA}^{DM}. Je podobná relácii R_{AA}^{MD} a predpokladáme, že sa výsledky ohodnotenia nebudú výrazne líšiť. Napriek tomu by sa odlišná meta cesta mala prejaviť vo výsledkoch ohodnotenia. Experimenty na tejto meta ceste sme taktiež vykonávali pre váženú aj binarizovanú maticu.

Pre vážené matice dopadli výsledky algoritmov PageRank (Obrázok 18) a HITS (Obrázok 19) podľa predpokladu - poradie ohodnotenia ostalo identické, ale hodnoty ohodnotenia sa líšia.

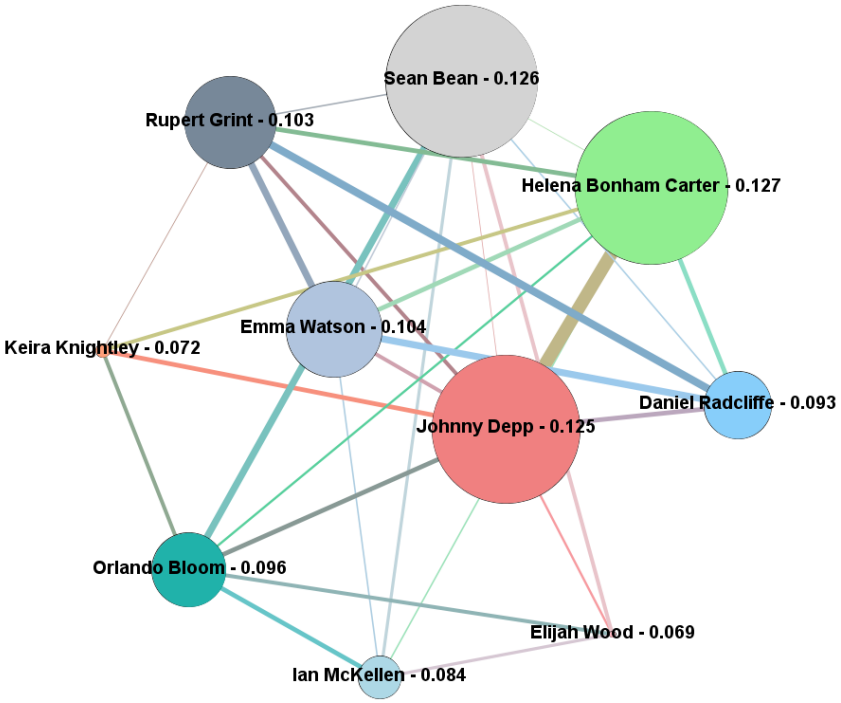


Obr. 16: Meta cesta ADMA, algoritmus PageRank, vážená matica, r_5

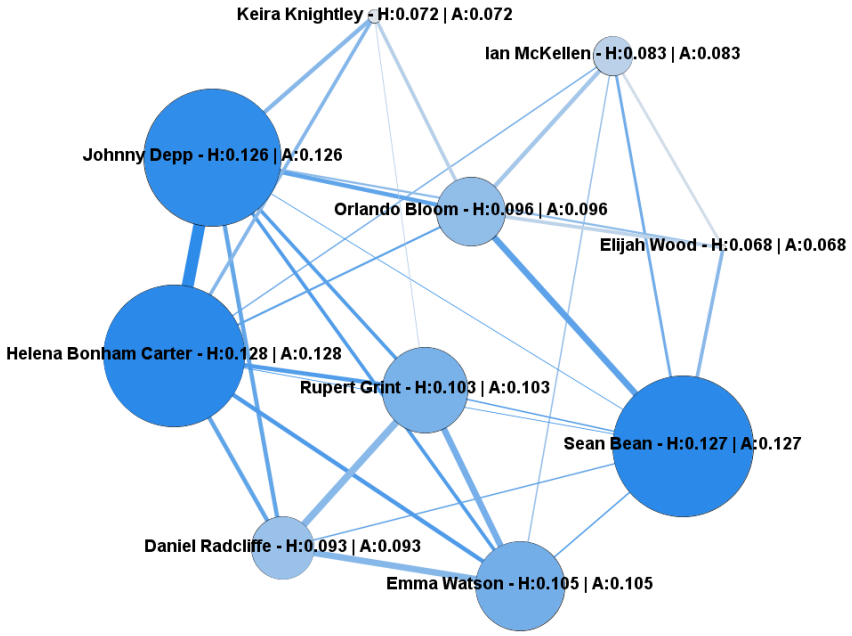


Obr. 17: Meta cesta ADMA, algoritmus HITS (k=10), vážená matica, r_6

V prípade binárnej matice sme úplne zanedbali počty prepojení, poradie a ohodnotenie algoritmami PageRank (Obrázok 16) a HITS (Obrázok 17) sú preto identické s meta cestou AMDA.



Obr. 18: Meta cesta ADMA, algoritmus PageRank, binárna matica, r_7



Obr. 19: Meta cesta ADMA, algoritmus HITS (k=10), binárna matica, r_8

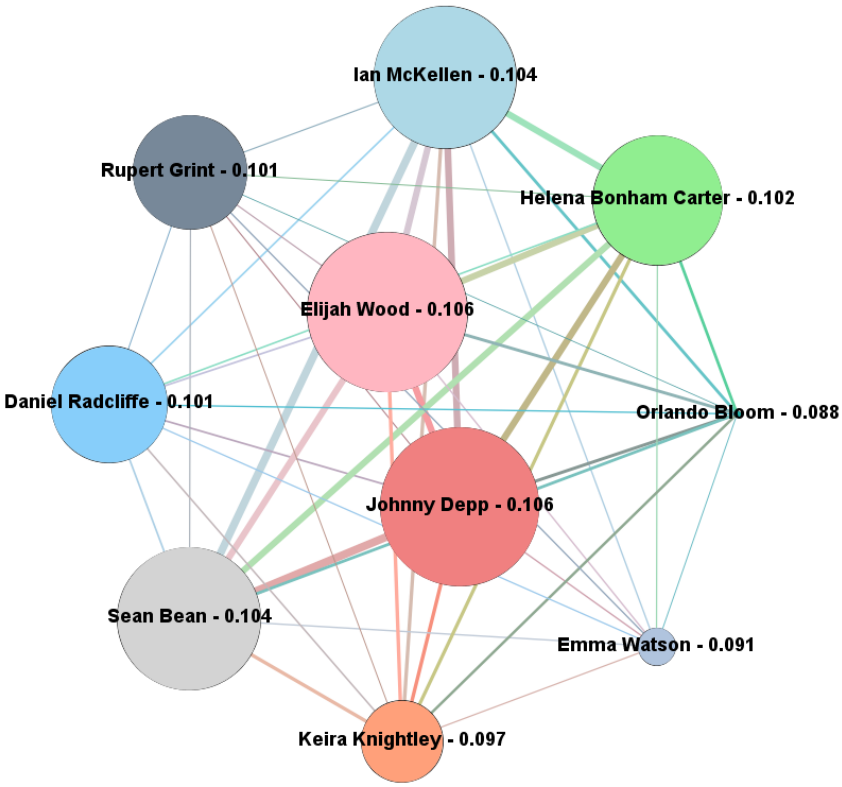
V prípade algoritmu Chain of Influencers (Tabuľka 5) sa poradie výsledkov líši, pretože kladie väčší dôraz na jednotlivé relácie meta cesty. Oproti meta ceste AMDA sa do popredia ohodnotenia dostala oktalógia Harry Potter a jej režiséri.

Tabuľka 5: Meta cesta ADMA, algoritmus Chain of Influencers, r_9

Actor	r_A	Director	r_D	Movie	r_M
Johnny Depp	0.27	Mike Newell	0.01	Corpse Bride	0.01
Helena B. Carter	0.22	David Yates	0.01	Pirates of the Caribbean: Dea...	0.010
Sean Bean	0.11	Gore Verbinski	0.01	Donnie Brasco	0.01
Orlando Bloom	0.08	Kenneth Branagh	0.01	Harry Potter and the Goblet...	0.01
Daniel Radcliffe	0.06	Tim Burton	0.01	Harry Potter and the Order o...	0.01
Ian McKellen	0.06	Mike Johnson	0.01	Harry Potter and the Half-Bl...	0.01
Emma Watson	0.06	James Bobin	0.01	Harry Potter and the Deathly...	0.01
Rupert Grint	0.06	Steve Kemsley	0.01	Harry Potter and the Deathly...	0.01
Elijah Wood	0.05	Joachim Rønning	0.01	Great Expectations	0.01
Keira Knightley	0.05	Espen Sandberg	0.01	Fantastic Beasts and Where t...	0.01

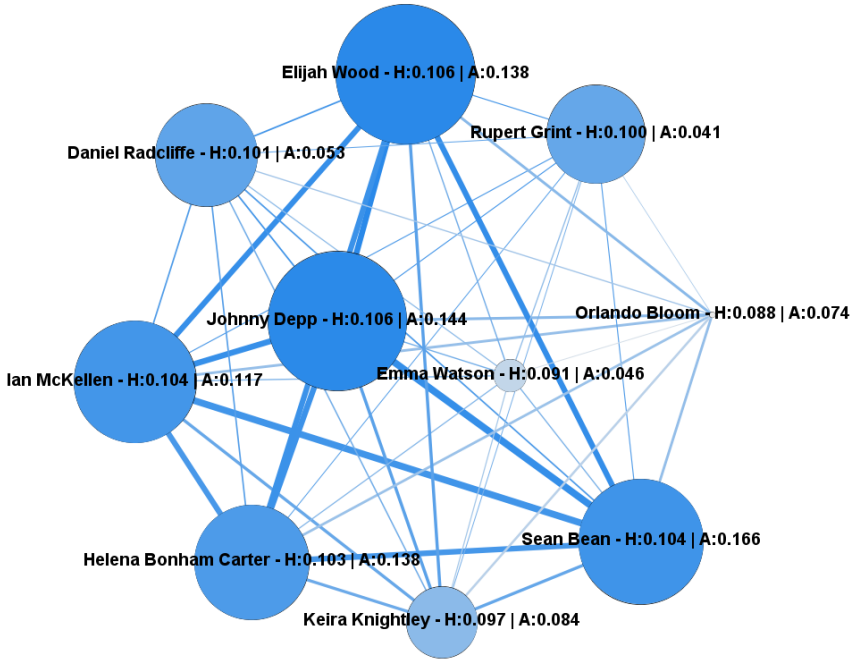
6.4 Meta cesta AGDA

Meta cesta AGDA reprezentuje reláciu hercov R_{AA}^{GD} ovplyvnenú filmovými žánrami a režisérmi. Pre túto meta cestu budeme analyzovať iba váženú maticu, pretože je plne prepojená a binarizáciou by vznikla symetrická matica.



Obr. 20: Meta cesta AGDA, algoritmus PageRank, vážená matica, r_{10}

V porovnaní s meta cestami AMDA a ADMA sa ohodnotenie váženej matice výrazne líši. Medzi ohodnoteniami jednotlivých hercov nie sú výrazné rozdiely. Herci tejto meta cesty spolu hrali vo filmoch rovnakých žánrov, výsledkom čoho je prirodzenejšie vyzerajúce ohodnotenie. Algoritmus PageRank (Obrázok 20) a HITS (Obrázok 21) opakovane dosahujú veľmi podobné výsledky.



Obr. 21: Meta cesta AGDA, algoritmus HITS (k=10), vážená matica, r_{11}

Výsledky algoritmu Chain of Influencers sú zobrazené v tabuľke 6. Všetky žánre boli ohodnotené rovnako. Ohodnotenie režisérov sa líši, to však kvôli zaokrúhleniu nie je možné vidieť. Vzhľadom na predošlé meta cesty sa množina režisérov rozšírila o nové mená, ktoré však lepšie reprezentujú túto meta cestu s ohľadom na žánre.

Tabuľka 6: Meta cesta ADGA, algoritmus Chain of Influencers, r_{12}

Actor	r_A	Genre	r_G	Director	r_D
Sean Bean	0.17	Crime	0.05	Robert Zemeckis	0.01
Johnny Depp	0.14	Drama	0.05	Kenneth Branagh	0.01
Elijah Wood	0.14	Romance	0.05	Michael Apted	0.01
Helena Bonham Carter	0.14	Thriller	0.05	Uli Edel	0.01
Ian McKellen	0.12	Mystery	0.05	Ron Howard	0.01
Keira Knightley	0.08	Comedy	0.05	Tim Burton	0.01
Orlando Bloom	0.07	Action	0.05	Ridley Scott	0.01
Daniel Radcliffe	0.05	Fantasy	0.05	Marc Forster	0.01
Emma Watson	0.05	Adventure	0.05	Roger Spottiswoode	0.01
Rupert Grint	0.04	Biography	0.05	Barry Levinson	0.01

Kvalitu a podobnosť ohodnotenia testovaných meta ciest vyhodnotíme pomocou Spearmanovho koeficientu ohodnotenia. Tento koeficient nadobúda hodnotu $< -1, 1 >$ podľa toho, aké podobné je zoradenie porovnávaných ohodnotení. V prípade, že je poradie ohodnotení podobné, koeficient sa blíži k 1. V opačnom prípade sa blíži k -1. Ak porovnáme navzájom všetky výsledky ohodnotení na testovaných meta cestách, môžeme podobnosť ohodnotení reprezentovať tabuľkou 7.

Tabuľka 7: Výsledky Spearmanovej korelácie meta ciest

	r_0	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}
r_0	1.00	0.77	1.00	0.67	0.64	1.00	0.64	1.00	0.49	0.77	0.02	0.02	0.21
r_1	0.77	1.00	0.77	0.93	0.82	0.77	0.82	0.77	0.77	1.00	0.25	0.25	0.47
r_2	1.00	0.77	1.00	0.67	0.64	1.00	0.64	1.00	0.49	0.77	0.02	0.02	0.21
r_3	0.67	0.93	0.67	1.00	0.96	0.67	0.96	0.67	0.92	0.93	0.16	0.16	0.52
r_4	0.64	0.82	0.64	0.96	1.00	0.64	1.00	0.64	0.95	0.82	0.09	0.09	0.53
r_5	1.00	0.77	1.00	0.67	0.64	1.00	0.64	1.00	0.49	0.77	0.02	0.02	0.21
r_6	0.64	0.82	0.64	0.96	1.00	0.64	1.00	0.64	0.95	0.82	0.09	0.09	0.53
r_7	1.00	0.77	1.00	0.67	0.64	1.00	0.64	1.00	0.49	0.77	0.02	0.02	0.21
r_8	0.49	0.77	0.49	0.92	0.95	0.49	0.95	0.49	1.00	0.77	0.28	0.28	0.72
r_9	0.77	1.00	0.77	0.93	0.82	0.77	0.82	0.77	0.77	1.00	0.25	0.25	0.47
r_{10}	0.02	0.25	0.02	0.16	0.09	0.02	0.09	0.02	0.28	0.25	1.00	1.00	0.78
r_{11}	0.02	0.25	0.02	0.16	0.09	0.02	0.09	0.02	0.28	0.25	1.00	1.00	0.78
r_{12}	0.21	0.47	0.21	0.52	0.53	0.21	0.53	0.21	0.72	0.47	0.78	0.78	1.00

Z tabuľky 7 vidíme, ktoré algoritmy ohodnotili meta cesty podobne a ktoré sa líšia. Tabuľka ukazuje, že rozličné meta cesty majú vplyv na výsledky ohodnotenia. Algoritmy PageRank a HITS dosahujú vo všetkých prípadoch navzájom porovnateľné výsledky, pretože pracujú len s výslednou maticou susednosti. Algoritmus Chain of Influencers skúma relácie meta ciest podrobnejšie a prináša tak odlišný spôsob ohodnotenia.

6.5 Meta cesta AMD

Meta cesta AMD relácie hercov a režisérov R^M_{AD} je na rozdiel od predošlých meta ciest reprezentovaná bipartitným grafom. Takúto meta cestu môžeme analyzovať algoritmom RankClus, ktorý pracuje s dvomi typmi objektov v sieti. Výstupom algoritmu (Tabuľka 8) sú zhluky hercov a ohodnotenie režisérov, s ktorými herci spolupracovali. Algorimus rozdelí hercov do disjunktných zhlukov a pre každý zhluk ohodnotí celú množinu režisérov. Pre prehľadnosť sme vybrali 10 režisérov s najvyšším ohodnotením v rámci zhluku. Algorimus je nedeterministický. Hercov rozdelí do zhlukov náhodne a stáva sa, že pri výpočte uviazne v lokálnom minime. Preto algorimus spustíme opakovane a vyberieme výsledok, ktorého súčty ohodnotení režisérov sú najvyššie.

Tabuľka 8: Meta cesta AMD, algoritmus RankClus (k=3)

Actor	Director	r_D
Johnny Depp	Tim Burton	0.09
Orlando Bloom	Gore Verbinski	0.07
Keira Knightley	Terry Gilliam	0.02
	Lasse Hallström	0.02
	Kevin Smith	0.02
	David Koepp	0.02
	Rob Marshall	0.02
	Joachim Rønning	0.02
	Espen Sandberg	0.02
	Yvan Attal	0.01
Helena Bonham Carter	David Yates	0.08
Daniel Radcliffe	Tim Burton	0.08
Rupert Grint	James Ivory	0.03
Emma Watson	Kenneth Branagh	0.02
	Trevor Nunn	0.02
	David Hare	0.02
	Tom Hooper	0.02
	Mike Newell	0.02
	Steve Kemsley	0.02
	Chris Columbus	0.02
Sean Bean	Tom Clegg	0.19
Ian McKellen	Peter Jackson	0.04
Elijah Wood	Chris Columbus	0.02
	Derek Jarman	0.02
	Phillip Noyce	0.02
	Terry Winsor	0.02
	Liam O Mochain	0.01
	Mike Figgis	0.01
	Ridley Scott	0.01
	Bruce Robinson	0.01

Algoritmus správne rozdelil hercov do zhlukov podľa toho, v akej filmovej sérii hrali. Ohodnotenie režisérov je konzistentné s predošlými meta cestami a priamo koreluje s hercami, ktorí hrali vo filmoch daného režiséra.

6.6 Meta cesta MAD

Meta cesta MAD relácie filmov a režisérov R^A_{MD} je taktiež reprezentovaná bipartitným grafom a na analýzu použijeme algoritmus RankClus. Množina filmov obsahuje 413 titulov, ktoré algoritmus rozdelil do 3 zhlukov. Zobrazenie všetkých filmov pomocou tabuľky je nepraktické, vyberieme len 10 titulov. Vybrané tituly nie sú nijakým spôsobom zoradené ani ohodnotené.

Pre každú množinu filmov algoritmus ohodnotil režisérov. Rozdelenie do zhlukov filmov a ohodnotenie režisérov (Tabuľka 9) korešponduje s filmami, ktoré režiséri režírovali.

6.7 Meta cesta DAM

Algoritmus RankClus (Tabuľka 10) rozdelil filmy relácie R^A_{DM} do 3 zhlukov a režisérov každého zhluku ohodnotil. Rozdelenie do zhlukov korešponduje s meta cestou MAD. Aplikovaním algoritmu na obidve meta cesty sme získali zhluky aj ohodnotenie pre všetky objekty. V tabuľke sme opäť z dôvodu veľkej množiny filmov nezobrazili všetky záznamy. Algoritmus RankClus na vybraných meta cestách vracia konzistentne uspokojivé výsledky.

Tabuľka 9: Meta cesta MAD, algoritmus RankClus (k=3)

Movie	Director	r_D
Harry Potter and the Sorcerer's Stone	Mike Newell	0.04
Harry Potter and the Chamber of Secrets	David Yates	0.04
Harry Potter and the Prisoner of Azkaban	Steve Kemsley	0.04
Harry Potter and the Goblet of Fire	Chris Columbus	0.04
Tom Felton Meets the Superfans	Alfonso Cuarón	0.04
The Tailor of Panama	Tom Felton	0.03
December Boys	Steve Paley	0.02
It's Christmas with Jonathan Ross	John Boorman	0.02
My Boy Jack	Rod Hardy	0.02
Kill Your Darlings	Paul McGuigan	0.02
...		
The Lord of the Rings: The Fellowship of the Ring	Peter Jackson	0.02
The Lord of the Rings: The Return of the King	Liam O Mochain	0.01
The Lord of the Rings: The Two Towers	Mike Figgis	0.01
The Book That Wrote Itself	Wolfgang Petersen	0.01
Troy	Ridley Scott	0.01
Henry VIII	Paul W.S. Anderson	0.01
Caravaggio	Pete Travis	0.01
Stormy Monday	Chris Columbus	0.01
War Requiem	Bruce Robinson	0.01
The Fifteen Streets	Derek Jarman	0.01
...		
Corpse Bride	Gore Verbinski	0.02
Charlie and the Chocolate Factory	Mike Newell	0.02
Sweeney Todd: The Demon Barber of Fleet Street	David Yates	0.02
Alice in Wonderland	Kenneth Branagh	0.02
Dark Shadows	Tim Burton	0.02
The Lone Ranger	Mike Johnson	0.02
Alice Through the Looking Glass	James Bobin	0.02
Pirates of the Caribbean: The Curse of the Black Pearl	Joachim Rønning	0.01
Pirates of the Caribbean: Dead Man's Chest	Espen Sandberg	0.01
Pirates of the Caribbean: At World's End	Yvan Attal	0.01
...		

Tabuľka 10: Meta cesta DAM, algoritmus RankClus (k=3)

Director	Movie	r_M
Peter Jackson	The Lord of the Rings: The Fellowship of the Ring	0.01
Liam O Mochain	The Lord of the Rings: The Return of the King	0.01
Mike Figgis	The Lord of the Rings: The Two Towers	0.01
Wolfgang Petersen	The Book That Wrote Itself	0.01
Ridley Scott	Troy	0.01
Paul W.S. Anderson	Henry VIII	0.01
Chris Columbus	Caravaggio	0.01
Pete Travis	Stormy Monday	0.01
Bruce Robinson	War Requiem	0.01
Derek Jarman	The Fifteen Streets	0.01
...		
Mike Newell	Corpse Bride	0.01
David Yates	Charlie and the Chocolate Factory	0.01
Gore Verbinski	Sweeney Todd: The Demon Barber of Fleet Street	0.01
Kenneth Branagh	Alice in Wonderland	0.01
Tim Burton	Dark Shadows	0.01
Mike Johnson	The Lone Ranger	0.01
James Bobin	Alice Through the Looking Glass	0.01
Joachim Rønning	Pirates of the Caribbean: The Curse of the Black Pearl	0.01
Espen Sandberg	Pirates of the Caribbean: Dead Man's Chest	0.01
Yvan Attal	Pirates of the Caribbean: At World's End	0.01
...		
Steve Kemsley	Harry Potter and the Order of the Phoenix	0.04
Alfonso Cuarón	Harry Potter and the Half-Blood Prince	0.04
Tom Felton	Harry Potter and the Deathly Hallows: Part 1	0.04
John Boorman	Harry Potter and the Deathly Hallows: Part 2	0.04
Rod Hardy	Harry Potter and the Deathly Hallows T4 Premiere Special	0.04
Paul McGuigan	Harry Potter and the Sorcerer's Stone	0.03
Alexandre Aja	Harry Potter and the Chamber of Secrets	0.03
Michael Dowse	Harry Potter and the Prisoner of Azkaban	0.03
Judd Apatow	Harry Potter and the Goblet of Fire	0.03
Greg McLean	Tom Felton Meets the Superfans	0.03
...		

7 Záver

Diplomová práca bola zameraná na problematiku analýzy heterogénnych sietí. Ukázali sme možnosti, ako tieto siete reprezentovať, príklady dátových sád reálneho sveta a analytické metódy, ktoré v tejto oblasti môžeme využiť. Preskúmali sme bázové sady a meta cesty, ktoré sa v heterogénnych sieťach nachádzajú. V rámci práce sme vytvorili databázu pre IMDb, ktorú sme použili ako zdroj dát pre implementované algoritmy.

Analýza sietí sa sústredila na ohodnotenie grafu. Vybrali sme niekoľko algoritmov, ktoré boli v minulosti použité na analýzu homogénnych sietí a aplikovali sme ich na projekciu heterogénnej siete. Taktiež sme implementovali algoritmy, ktoré pracujú priamo s heterogénnymi reláciami.

Spojením týchto častí sme vytvorili framework na analýzu heterogénnych sietí. Databázová vrstva bola navrhnutá s dôrazom na flexibilitu, čo umožňuje zámenu domény, s ktorou framework pracuje. Užívateľské rozhranie umožňuje jednoduchý prístup k filtrovaniu načítavaných dát z databázy, ktoré následne analyzujeme. Načítané záznamy prevádzame na váženú maticu pomocou meta ciest, ktoré zavádzajú do siete sémantiku. Takúto sieť následne analyzujeme implementovanými algoritmami a výsledky exportujeme do formátu grafu `GEXF`. Ten umožní sieť vizualizovať cez nástroj Gephi. Alternatívou je dátový formát `TSV`.

Analýzu dátovej sady IMDb sme vykonali experimentálne. Vybrali sme vhodnú bázovú sadu a meta cesty, na ktoré sme aplikovali algoritmy. Výsledky experimentov sme vizualizovali formou grafov, tabuliek a slovným popisom. Predpokladali sme, že rozličné meta cesty majú vplyv na ohodnotenie vrcholov grafu. Ukázali sme, že sémantika rôznych meta ciest naozaj ovplyvňuje výsledné ohodnotenie vrcholov. Všetky experimenty sú súčasťou prílohy.

Framework ponúka možnosti pokračovania tejto práce. Môžeme ho jednoducho rozšíriť o ďalšie vhodné algoritmy. Zmenou databázovych modelov ho vieme prispôsobiť inej dátovej sade.

8 Literatúra

[1] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun and Philip S. Yu. *A Survey of Heterogeneous Information Network Analysis.* In: IEEE Transactions on Knowledge and Data Engineering, 2017, p. 17-37

[2] Erheng Zhong, Wei Fang, Yin Zhu and Qiang Yang. *Modeling the dynamics of composite social networks.* In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, 2013, p. 937-945

[3] Mohsen Jamali and Laks Lakshmanan. *HeteroMF: recommendation in heterogeneous information networks using context dependent factor models.* In: Proceedings of the 22nd international conference on World Wide Web, 2013, p. 643-654

[4] Bo Long, Zhongfei Mark Zhang and Philip S. Yu. *Co-clustering by block value decomposition.* In: Proceedings of the 11th ACM SIGKDD international conference on Knowledge discovery and data mining, 2005, p. 635-640

[5] Bo Long, Zhongfei Mark Zhang, Xiaoyun Wu and Philip S. Yu. *Unsupervised learning on K-partite graphs.* In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006, p. 317-326

[6] Honglei Zhuang, Jing Zhang, George Brova, Jie Tang, Hasan Cam, Xifeng Yan and Jiawei Han. *Mining Query-Based Subnetwork Outliers in Heterogeneous Information Networks.* In: 2014 IEEE International Conference on Data Mining, 2014, p. 1127-1132

[7] Xiangnan Kong, Bokai Cao and Philip S. Yu. *Multi-label classification by mining label and instance correlations from heterogeneous information networks.* In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, 2013, p. 614-622

[8] A. Singhal. *Introducing the knowledge graph: things, not strings.* Official Google Blog, 2012

[9] Anil K. Jain. *Data clustering: 50 years beyond k-means.* In: Pattern Recognition Letters, 2010, p. 651-666

[10] Jianbo Shi and Jitendra Malik. *Normalized cuts and image segmentation.* In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, p. 888-905

[11] Mark Newman and Michelle Girvan. *Finding and evaluating community structure in networks.* In: Physical review E, vol. 69, no. 2, 2004

[12] David Liben-Nowell and Jon Kleinberg. *The link-prediction problem for social networks.* Journal of the American society for information science and technology, 2003, p. 1019-1031

[13] Alexandrin Popescul and Lyle H. Ungar. *Statistical relational learning for link prediction.* In: Workshop on Learning Statistical Models from Relational Data, 2003

[14] Nathan Srebro and Tommi Jaakkola. *Weighted low-rank approximations.* In: Proceedings of the Twentieth International Conference, 2003, p. 720-727

[15] Chuan Shi, Chong Zhou, Xiangnan Kong, Philip S. Yu, Gang Liu and Bai Wang. *Heterecom: a semantic-based recommendation system in heterogeneous networks.* In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012, p. 1552-1555

[16] Chuan Shi, Zhiqiang Zhang, Ping Luo, Philip S. Yu, Yading Yue and Bin Wu. *Semantic path based personalized recommendation on weighted heterogeneous information networks.* In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015, p. 453-462

[17] Jiawei Zhang and Philip S. Yu. *Integrated anchor and social link predictions across social networks.* In: Proceedings of the 24th International Conference on Artificial Intelligence, 2015, p. 2125-2131

[18] Lawrence Page, Sergey Brin, Rajeev Motwani and Terry Winograd. *The PageRank Citation Ranking: Bringing Order to the Web.* 1998

[19] Erjia Yan, Ying Ding, Rajeev Motwani and Terry Winograd. *Discovering author impact: A PageRank perspective.* In: Information Processing and Management: an International Journal, 2011, p. 125-134

[20] Jon M. Kleinberg. *Authoritative Sources in a Hyperlinked Environment.* In: Journal of the ACM (JACM), 46.5, 1999, p. 604-632

[21] Pavla Dráždilová, Jan Konečný, and Miloš Kudělka. *Chain of Influencers: Multipartite Intra-community Ranking.* In: Computing and Combinatorics: 23rd International Conference, 2017, p. 603-614

[22] Glen Jeh and Jennifer Widom. *SimRank: A Measure of Structural-Context Similarity.* In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, 2002, p. 538-543

[23] Xiaoxin Yin, Jiawei Han and Philip S. Yu. *LinkClus: efficient clustering via heterogeneous semantic links.* In: Proceedings of the 32nd international conference on Very large data bases, 2006, p. 427-438

[24] Yizhou Sun, Jiawei Han, Peixiang Zhao, Zhijun Yin, Hong Cheng and Tianyi Wu. *RankClus: Integrating Clustering with Ranking for Heterogeneous Information Network Analysis.* In: 12th International Conference on Extending Database Technology, 2009, p. 565-576

[25] Arthur Dempster, Natalie Laird and Donald B. Rubin. *Maximum Likelihood From Incomplete Data Via The EM algorithm*. In: Journal of the Royal Statistical Society: Series B (Statistical Methodology), 1977, p. 1-38

9 Príloha na CD/DVD

Štruktúra príloh:

- /code - Zdrojový kód programu
- /document - Tento dokument vo formáte PDF
- /experiments - Výsledky experimentov
- /mysql - Databáza IMDb